

PENGEMBANGAN *DASHBOARD* SISTEM PERSEBARAN CYBERBULLYING PADA MEDIA SOSIAL TWITTER

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Aditya Wisnu Jati Kusumo

NIM: 145150201111146



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PENGEMBANGAN DASHBOARD SISTEM PERSEBARAN CYBERBULLYING PADA
MEDIA SOSIAL TWITTER

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Aditya Wisnu Jati Kusumo
NIM: 145150201111146

Skripsi ini telah diuji dan dinyatakan lulus pada
27 April 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Fajar Pradana, S.ST, M.Eng
NIP: 19871121 201504 1 004

Dosen Pembimbing II



Bayu Priyambadha, S.Kom, M.Kom
NIP: 19820909 200812 1 004

Mengetahui

Ketua Jurusan Teknik Informatika



Kri Astoro Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 4 April 2018



Aditya Wisnu Jati Kusumo

NIM: 145150201111146

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT karena berkat karuniaNya, penulis dapat menyelesaikan penulisan skripsi ini sebagai syarat untuk memperoleh gelar sarjana pada Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.

Penulis menyadari dalam pengerjaan skripsi ini, masih terdapat banyak kekurangan baik dari segi teknis maupun non teknis, hal ini mengingat keterbatasan pengetahuan yang dimiliki dan kekhilafan penulis sebagai manusia. Besar harapan penulis melalui penulisan skripsi ini dapat memberikan sumbangsih terhadap perkembangan ilmu pengetahuan.

Dalam penyusunan skripsi ini, penulis mendapat banyak bantuan dari berbagai pihak terlepas dari kekurangan penulis, baik berupa materil, ilmu, maupun spiritual. Dalam kesempatan ini penulis ingin mengucapkan terima kasih kepada :

1. Kedua orang tua, adik, serta seluruh keluarga besar saya yang selalu memberikan doa, dukungan, serta motivasi kepada saya.
2. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya.
4. Bapak Fajar Pradana, S.ST, M.Eng selaku Dosen Pembimbing I, yang selalu membimbing, mengarahkan, dan memberikan ilmu yang bermanfaat kepada penulis dalam menyelesaikan penulisan skripsi dengan baik.
5. Bapak Bayu Priyambadha, S.Kom, M.Kom selaku Dosen Pembimbing II, yang selalu membimbing, mengarahkan, dan memberikan ilmu yang bermanfaat kepada penulis dalam menyelesaikan penulisan skripsi dengan baik.
6. Seluruh dosen beserta staff di Fakultas Ilmu komputer Universitas Brawijaya yang selalu memberikan ilmu yang bermanfaat bagi penulis.
7. Seluruh teman-teman yang selalu memberikan ilmu, dukungan, beserta doa yang tak henti-hentinya.
8. Seluruh praktikan saya yang selalu membagikan dukungan, doa, serta perhatiannya.
9. Seluruh rekan mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya yang selalu membantu penulis selama masa studi hingga terselesaikannya skripsi ini.
10. Seluruh pihak yang tidak dapat penulis sebutkan satu persatu yang telah membantu terselesaikannya penulisan skripsi ini.

Penulis menyadari dalam pengerjaan skripsi ini masih terdapat banyak kekurangan baik dari segi teknis maupun non teknis, hal ini mengingat keterbatasan pengetahuan yang dimiliki dan kekhilafan penulis sebagai manusia. Besar harapan penulis melalui penulisan skripsi ini dapat memberikan sumbangsih terhadap perkembangan ilmu pengetahuan serta dapat bermanfaat bagi seluruh pihak yang memerlukan.

Malang, 4 April 2018

Penulis

Aditya.wisnu373@gmail.com



ABSTRAK

Aditya Wisnu Jati Kusumo, Pengembangan *Dashboard* Sistem Persebaran *Cyberbullying* Pada Media Sosial Twitter

Pembimbing: Fajar Pradana, S.ST, M.Eng dan Bayu Priyambadha, S.Kom, M.Kom

Penelitian di bidang teknologi komunikasi saat ini telah berkembang pesat. Salah satu bukti hasil dari perkembangan teknologi ialah penggunaan internet sebagai media komunikasi. Disisi lain perkembangan teknologi dapat menimbulkan dampak negatif yang berbahaya, seperti *cyberbullying*. *Cyberbullying* merupakan segala bentuk kekerasan yang dilakukan di dalam dunia maya yang dapat menimbulkan gangguan kesehatan bagi korbannya, baik fisik maupun psikologis. Kurangnya perhatian mengenai masalah *cyberbullying*, serta tidak tersedianya data mengenai *cyberbullying* untuk menggerakkan tindakan preventif menjadi salah satu masalah dalam menanggulangi kasus *cyberbullying*. Penelitian ini dilakukan dengan tujuan untuk menyita perhatian dari pemerintah melalui kementerian komunikasi dan informatika mengenai kasus *cyberbullying* serta dapat dijadikan dasar di dalam menggerakkan tindakan-tindakan preventif, dengan menyediakan data mengenai *cyberbullying* dan memvisualisasikan informasi tersebut kedalam sebuah *dashboard* sistem. Tahapan dalam penelitian ini dimulai dari studi literatur, pengumpulan informasi, analisis kebutuhan, perancangan, implementasi, serta pengujian. Pada penelitian ini terdapat beberapa proses pengembangan perangkat lunak, yang dilakukan dengan model pengembangan *waterfall*. Sistem yang dikembangkan dalam penelitian ini merupakan sistem berbasis web yang dapat menampilkan persentase *cyberbullying* pada suatu kota dengan data didapatkan dari *tweet* pada Twitter yang telah diklasifikasikan dengan menggunakan analisis sentimen. *Dashboard* yang dikembangkan dalam penelitian ini dapat mengintegrasikan antara *framework* yang digunakan, *library* API Twitter sebagai pengumpul data, serta *library* analisis sentimen untuk mengklasifikasikan data. Strategi pengujian pada penelitian ini dilakukan dengan menggunakan pengujian unit, pengujian integrasi, serta pengujian validasi, dengan memberikan nilai valid untuk seluruh kasus uji yang artinya sistem yang dibangun dapat berjalan sesuai tujuan dan bebas dari adanya kegagalan.

Kata kunci: *internet, cyberbullying, dashboard, API Twitter, analisis sentimen*

ABSTRACT

Research in the field of communications technology today has grown rapidly. This is indicated by the use of the internet as a medium of communication. The development of technology can cause harmful effects, such as cyberbullying. Cyberbullying is any form of violence committed in the virtual world that can cause health problems for victims, such as physical and psychological problems. Lack of concern about cyberbullying issues, and the unavailability of cyberbullying data to actuate preventive action is the problems to overcome cyberbullying issues. The study was conducted with the aim of seizing the attention of the government through the communications and informatics ministries on cyberbullying cases and can serve as a basis in mobilizing preventive actions, by providing data on cyberbullying and visualizing the information into a system dashboard. Stages in this study started from literature studies, information gathering, requirement analysis, design, implementation, and testing. In this research, there are several software development processes, which is done with waterfall development model. The developed system in this study is a web-based system that can display the percentage of cyberbullying in a location with data obtained from tweets on Twitter that have been classified using sentiment analysis. The developed system in this research integrates between the framework used, the Twitter API library as data collector, and the library of sentiment analysis to classify the data. The testing strategy in this research is done by unit testing, integration testing, and validation testing, with the result all valid value for all test cases which means that the built system can run as intended and free from failure.

Keyword: *internet, cyberbullying, dashboard, Twitter API, sentiment analysis*

DAFTAR ISI

| | |
|---------------------------------------------------|-------------------------------------|
| PENGESAHAN..... | Error! Bookmark not defined. |
| PERNYATAAN ORISINALITAS..... | Error! Bookmark not defined. |
| KATA PENGANTAR..... | 4 |
| ABSTRAK | 6 |
| ABSTRACT..... | 7 |
| DAFTAR ISI..... | 8 |
| DAFTAR TABEL | 12 |
| DAFTAR GAMBAR | 15 |
| DAFTAR LAMPIRAN | Error! Bookmark not defined. |
| BAB 1 PENDAHULUAN..... | Error! Bookmark not defined. |
| 1.1 Latar belakang..... | Error! Bookmark not defined. |
| 1.2 Rumusan masalah..... | Error! Bookmark not defined. |
| 1.3 Tujuan..... | Error! Bookmark not defined. |
| 1.4 Manfaat..... | Error! Bookmark not defined. |
| 1.5 Batasan masalah | Error! Bookmark not defined. |
| 1.6 Sistematika pembahasan | Error! Bookmark not defined. |
| BAB 2 LANDASAN KEPUSTAKAAN..... | Error! Bookmark not defined. |
| 2.1 Kajian Pustaka | Error! Bookmark not defined. |
| 2.2 <i>Dashboard</i> | Error! Bookmark not defined. |
| 2.3 <i>Cyberbullying</i> | Error! Bookmark not defined. |
| 2.3.1 <i>Dampak Cyberbullying</i> | Error! Bookmark not defined. |
| 2.4 Model Pengembangan <i>Waterfall</i> | Error! Bookmark not defined. |
| 2.5 <i>Unified Modelling Language (UML)</i> | Error! Bookmark not defined. |
| 2.6 Twitter | Error! Bookmark not defined. |
| 2.6.1 Twitter API | Error! Bookmark not defined. |
| 2.7 Google Maps | Error! Bookmark not defined. |
| 2.7.1 Google Maps API | Error! Bookmark not defined. |
| 2.8 Analisis Sentimen | Error! Bookmark not defined. |
| BAB 3 METODOLOGI..... | Error! Bookmark not defined. |
| 3.1 Studi Literatur | Error! Bookmark not defined. |
| 3.2 Pengumpulan Informasi..... | Error! Bookmark not defined. |

| | |
|--------------------------------|-------------------------------------|
| 3.3 Analisis Kebutuhan..... | Error! Bookmark not defined. |
| 3.4 Perancangan..... | Error! Bookmark not defined. |
| 3.5 Implementasi | Error! Bookmark not defined. |
| 3.6 Pengujian..... | Error! Bookmark not defined. |
| 3.6.1 Pengujian Unit | Error! Bookmark not defined. |
| 3.6.2 Pengujian Integrasi..... | Error! Bookmark not defined. |
| 3.6.3 Pengujian Validasi | Error! Bookmark not defined. |

BAB 4 ANALISIS KEBUTUHAN.....Error! Bookmark not defined.

| | |
|------------------------------------------|-------------------------------------|
| 4.1 Gambaran Umum Sistem | Error! Bookmark not defined. |
| 4.2 Identifikasi Aktor | Error! Bookmark not defined. |
| 4.3 Daftar Kebutuhan Fungsional | Error! Bookmark not defined. |
| 4.4 Daftar Kebutuhan Non Fungsional..... | Error! Bookmark not defined. |
| 4.5 Pemodelan Kebutuhan..... | Error! Bookmark not defined. |
| 4.5.1 <i>Use Case</i> Diagram..... | Error! Bookmark not defined. |
| 4.5.2 <i>Use Case Scenario</i> | Error! Bookmark not defined. |

BAB 5 PERANCANGAN.....Error! Bookmark not defined.

| | |
|-----------------------------------------------------------|-------------------------------------|
| 5.1 Perancangan <i>Sequence</i> diagram..... | Error! Bookmark not defined. |
| 5.1.1 Mengambil Data..... | Error! Bookmark not defined. |
| 5.1.2 Mengambil Data Dengan Form Pencarian..... | Error! Bookmark not defined. |
| 5.1.3 Mengambil Data Pada Kota Yang Tersimpan..... | Error! Bookmark not defined. |
| 5.1.4 Melihat Persentase | Error! Bookmark not defined. |
| 5.1.5 Melihat Detail Data | Error! Bookmark not defined. |
| 5.1.6 Melihat Detail Data Tanggal Pilihan..... | Error! Bookmark not defined. |
| 5.1.7 Melihat Detail Data Hari Ini..... | Error! Bookmark not defined. |
| 5.1.8 Melihat Detail Data Kemarin..... | Error! Bookmark not defined. |
| 5.1.9 <i>Export</i> Data ke Excel..... | Error! Bookmark not defined. |
| 5.1.10 <i>Export</i> Data Ke Excel Pada Kota Pilihan..... | Error! Bookmark not defined. |
| 5.1.11 Melihat Waktu Ujaran Negatif Terbanyak..... | Error! Bookmark not defined. |
| 5.1.12 Melihat Kota Ujaran Negatif Terbanyak..... | Error! Bookmark not defined. |

| | |
|--------------------------------------------------------------------------------------------------|------------------------------|
| 5.1.13 Menghentikan Pengambilan Data | Error! Bookmark not defined. |
| 5.2 Perancangan <i>Class Diagram</i> | Error! Bookmark not defined. |
| 5.3 Perancangan Komponen | Error! Bookmark not defined. |
| 5.3.1 Algoritme Mengambil Data | Error! Bookmark not defined. |
| 5.3.2 Algoritme Mencari Persentase Ujaran Negatif | Error! Bookmark not defined. |
| 5.3.3 Algoritme Mengambil Data 1 Hari Lalu | Error! Bookmark not defined. |
| 5.4 Perancangan Basis Data | Error! Bookmark not defined. |
| 5.5 Perancangan Antar Muka | Error! Bookmark not defined. |
| 5.5.1 Halaman Home | Error! Bookmark not defined. |
| BAB 6 IMPLEMENTASI | Error! Bookmark not defined. |
| 6.1 Spesifikasi Perangkat Keras | Error! Bookmark not defined. |
| 6.2 Spesifikasi Perangkat Lunak | Error! Bookmark not defined. |
| 6.3 Batasan Implementasi | Error! Bookmark not defined. |
| 6.4 Implementasi Kode Program | Error! Bookmark not defined. |
| 6.4.1 Kode Program Mengambil Data | Error! Bookmark not defined. |
| 6.4.2 Kode Program Mencari Persentase Ujaran Negatif | Error! Bookmark not defined. |
| 6.4.3 Kode Program Mengambil Data 1 Hari Lalu | Error! Bookmark not defined. |
| 6.5 Implementasi Basis Data | Error! Bookmark not defined. |
| 6.6 Implementasi Antar Muka | Error! Bookmark not defined. |
| 6.6.1 Halaman Home | Error! Bookmark not defined. |
| BAB 7 PENGUJIAN | Error! Bookmark not defined. |
| 7.1 Pengujian Unit | Error! Bookmark not defined. |
| 7.1.1 Pengujian Unit Kelas <i>data_controller</i> untuk operasi <i>getSearhCityTwit(Request)</i> | Error! Bookmark not defined. |
| 7.1.2 Pengujian Unit Kelas <i>data_controller</i> untuk operasi <i>getPercentageNeg(Request)</i> | Error! Bookmark not defined. |
| 7.1.3 Pengujian Unit Kelas <i>data_controller</i> untuk operasi <i>getTwitYesterday(Request)</i> | Error! Bookmark not defined. |
| 7.2 Pengujian Integrasi | Error! Bookmark not defined. |
| 7.2.1 Operasi <i>decision(String)</i> | Error! Bookmark not defined. |

| | |
|--------------------------------------|-------------------------------------|
| 7.2.2 Operasi score() | Error! Bookmark not defined. |
| 7.3 Pengujian Validasi | Error! Bookmark not defined. |
| 7.3.1 Kebutuhan Fungsional..... | Error! Bookmark not defined. |
| 7.3.2 Kebutuhan Non Fungsional | Error! Bookmark not defined. |
| BAB 8 PENUTUP | Error! Bookmark not defined. |
| 8.1 Kesimpulan..... | Error! Bookmark not defined. |
| 8.2 Saran | Error! Bookmark not defined. |
| DAFTAR PUSTAKA | Error! Bookmark not defined. |



DAFTAR TABEL

| | |
|--------------------------------------------------------------------------------------|-------------------------------------|
| Tabel 2.1 Tabel Komponen <i>Sequence</i> Diagram..... | Error! Bookmark not defined. |
| Tabel 2.2 Tabel Komponen <i>Class</i> Diagram..... | Error! Bookmark not defined. |
| Tabel 4.1 Tabel Identifikasi Aktor..... | Error! Bookmark not defined. |
| Tabel 4.2 Tabel Kebutuhan Non Fungsional Sistem. | Error! Bookmark not defined. |
| Tabel 4.3 <i>Use Case Scenario</i> Mengambil Data | Error! Bookmark not defined. |
| Tabel 4.4 <i>Use Case Scenario</i> Mengambil Data Dengan Form Pencarian | Error! Bookmark not defined. |
| Tabel 4.5 <i>Use Case Scenario</i> Mengambil Data Pada Kota Yang Tersimpan.... | Error! Bookmark not defined. |
| Tabel 4.6 <i>Use Case Scenario</i> Melihat Persentase | Error! Bookmark not defined. |
| Tabel 4.7 <i>Use Case Scenario</i> Melihat Detail Data | Error! Bookmark not defined. |
| Tabel 4.8 <i>Use Case Scenario</i> Melihat detail data tanggal pilihan | Error! Bookmark not defined. |
| Tabel 4.9 <i>Use Case Scenario</i> Melihat Detail Data Hari ini | Error! Bookmark not defined. |
| Tabel 4.10 <i>Use Case Scenario</i> Melihat Detail Data Kemarin | Error! Bookmark not defined. |
| Tabel 4.11 <i>Use Case Scenario</i> Export Data ke Excel. | Error! Bookmark not defined. |
| Tabel 4.12 <i>Use Case Scenario</i> Export Data Ke Excel Pada Kota Pilihan..... | Error! Bookmark not defined. |
| Tabel 4.13 <i>Use Case Scenario</i> Menampilkan Waktu Ujaran Negatif Terbanyak | Error! Bookmark not defined. |
| Tabel 4.14 <i>Use Case Scenario</i> Melihat Kota Ujaran Negatif Terbanyak | Error! Bookmark not defined. |
| Tabel 4.15 <i>Use Case Scenario</i> Menghentikan Pengambilan Data | Error! Bookmark not defined. |
| Tabel 5.1 Algoritme Mengambil Data | Error! Bookmark not defined. |
| Tabel 5.2 Algoritme Mencari Persentase Ujaran Negatif | Error! Bookmark not defined. |
| Tabel 5.3 Algoritme Mengambil Data 1 Hari Lalu.... | Error! Bookmark not defined. |
| Tabel 6.1 Spesifikasi Perangkat Keras | Error! Bookmark not defined. |
| Tabel 6.2 Spesifikasi Perangkat Lunak | Error! Bookmark not defined. |
| Tabel 6.3 Kode Program Mengambil Data | Error! Bookmark not defined. |

Tabel 6.4 Kode Program Mencari Persentase Ujaran Negatif **Error! Bookmark not defined.**

Tabel 6.5 Kode Program Mengambil Data 1 Hari Lalu **Error! Bookmark not defined.**

Tabel 7.1 Pseudocode Pengujian Unit Operasi getSearchCityTwit(Request) . **Error! Bookmark not defined.**

Tabel 7.2 Pengujian Unit Kelas data_controller untuk operasi getSearchCityTwit (Request) **Error! Bookmark not defined.**

Tabel 7.3 Pseudocode Pengujian Unit Operasi getPercentageNeg(Request) **Error! Bookmark not defined.**

Tabel 7.4 Pengujian Unit Kelas data_controller untuk operasi getPercentageNeg (Request) **Error! Bookmark not defined.**

Tabel 7.5 Pseudocode Pengujian Unit Operasi getTwitYesterday(Request) .. **Error! Bookmark not defined.**

Tabel 7.6 Pengujian Unit Kelas data_controller untuk operasi getTwitYesterday(Request) **Error! Bookmark not defined.**

Tabel 7.7 Pseudocode Operasi decision(String) **Error! Bookmark not defined.**

Tabel 7.8 Pengujian Integrasi Operasi decision(String) **Error! Bookmark not defined.**

Tabel 7.9 Pseudocode Operasi score() **Error! Bookmark not defined.**

Tabel 7.10 Pengujian Integrasi Operasi score() **Error! Bookmark not defined.**

Tabel 7.11 Kasus Uji Mengambil Data..... **Error! Bookmark not defined.**

Tabel 7.12 Kasus Uji Mengambil Data Dengan Form Pencarian **Error! Bookmark not defined.**

Tabel 7.13 Kasus Uji Mengambil Data Pada Kota Yang Tersimpan **Error! Bookmark not defined.**

Tabel 7.14 Kasus Uji Melihat Persentase **Error! Bookmark not defined.**

Tabel 7.15 Kasus Uji Melihat Detail Data **Error! Bookmark not defined.**

Tabel 7.16 Kasus Uji Melihat Detail Data Tanggal Pilihan **Error! Bookmark not defined.**

Tabel 7.17 Kasus Uji Lihat Melihat Detail Data Hari Ini **Error! Bookmark not defined.**

Tabel 7.18 Kasus Uji Melihat Detail Data Kemarin... **Error! Bookmark not defined.**

Tabel 7.19 Kasus Uji *Export* Data ke Excel..... **Error! Bookmark not defined.**

Tabel 7.20 Kasus Uji *Export Data* Ke Excel Pada Kota Pilihan **Error! Bookmark not defined.**

Tabel 7.21 Kasus Uji Melihat waktu Ujaran Negatif Terbanyak **Error! Bookmark not defined.**

Tabel 7.22 Kasus Uji Melihat Kota Ujaran Negatif Terbanyak **Error! Bookmark not defined.**

Tabel 7.23 Kasus Uji Menghentikan Pengambilan Data **Error! Bookmark not defined.**

Tabel 7.24 Pengujian Validasi *Compability* **Error! Bookmark not defined.**



DAFTAR GAMBAR

- Gambar 2.1 Contoh Kasus *Cyberbullying* di Indonesia **Error! Bookmark not defined.**
- Gambar 2.2 Contoh Kasus *Cyberbullying* di Indonesia Pada Media Sosial Twitter **Error! Bookmark not defined.**
- Gambar 2.3 Contoh Kasus *Cyberbullying* di Indonesia Pada Media Sosial Twitter **Error! Bookmark not defined.**
- Gambar 2.4 Model Pengembangan *Waterfall* **Error! Bookmark not defined.**
- Gambar 2.5 *Use Case* Diagram Sumber : Fowler (2004) **Error! Bookmark not defined.**
- Gambar 2.6 Contoh *Sequence* Diagram Sumber : Arief (2012) **Error! Bookmark not defined.**
- Gambar 2.7 Alur Kerja *Library* Sentimen Analisis **Error! Bookmark not defined.**
- Gambar 2.8 Alur Integrasi *Library* Analisis Sentimen dan API Twitter **Error! Bookmark not defined.**
- Gambar 3.1 Diagram Alur Kerja **Error! Bookmark not defined.**
- Gambar 4.1 *Use Case* Diagram **Error! Bookmark not defined.**
- Gambar 5.1 *Sequence* Diagram Mengambil Data **Error! Bookmark not defined.**
- Gambar 5.2 *Sequence* Diagram Mengambil Data Dengan Form Pencarian ... **Error! Bookmark not defined.**
- Gambar 5.3 *Sequence* Diagram Mengambil Data Pada Kota Yang Tersimpan **Error! Bookmark not defined.**
- Gambar 5.4 *Sequence* Diagram Melihat Persentase **Error! Bookmark not defined.**
- Gambar 5.5 *Sequence* Diagram Melihat Detail Data **Error! Bookmark not defined.**
- Gambar 5.6 *Sequence* Diagram Melihat Detail Data Tanggal Pilihan **Error! Bookmark not defined.**
- Gambar 5.7 *Sequence* Diagram Melihat Detail Data Hari Ini **Error! Bookmark not defined.**
- Gambar 5.8 *Sequence* Diagram Melihat Detail Data Kemarin **Error! Bookmark not defined.**
- Gambar 5.9 *Sequence* Diagram *Export* Data Ke Excel **Error! Bookmark not defined.**
- Gambar 5.10 *Sequence* Diagram *Export* Data Ke Excel Pada Kota Pilihan **Error! Bookmark not defined.**
- Gambar 5.11 *Sequence* Diagram Melihat Waktu Ujaran Negatif Terbanyak . **Error! Bookmark not defined.**

Gambar 5.12 *Sequence Diagram* Melihat Kota Ujaran Negatif Terbanyak..... **Error! Bookmark not defined.**

Gambar 5.13 *Sequence Diagram* Menghentikan Pengambilan Data..... **Error! Bookmark not defined.**

Gambar 5.14 *Class Diagram***Error! Bookmark not defined.**

Gambar 5.15 *Conceptual Data Model* Sistem *Dashboard Monitoring Cyberbullying***Error! Bookmark not defined.**

Gambar 5.16 Halaman Home**Error! Bookmark not defined.**

Gambar 6.1 *Physical Data Model* Sistem *Dashboard Monitoring Cyberbullying***Error! Bookmark not defined.**

Gambar 6.2 Implementasi Antar Muka Halaman Home**Error! Bookmark not defined.**

Gambar 7.1 Flowgraph Pengujian Unit Operasi *getSearchCityTwit(Request)* **Error! Bookmark not defined.**

Gambar 7.2 Flowgraph Pengujian Unit Operasi *getPercentageNeg(Request)***Error! Bookmark not defined.**

Gambar 7.3 Flowgraph Pengujian Unit Operasi *getTwitKota(Request)* **Error! Bookmark not defined.**

Gambar 7.4 Flowgraph Pengujian Integrasi Operasi *decision(String)* **Error! Bookmark not defined.**

Gambar 7.5 Flowgraph Operasi *score()***Error! Bookmark not defined.**

1.1 Sistematika pembahasan

Untuk mencapai tujuan yang diharapkan, maka sistematika penulisan yang akan disusun dalam penelitian ini adalah sebagai berikut:

Bab I Pendahuluan

Pada bab ini akan membahas tentang latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika penulisan yang berkaitan dengan penelitian yang akan dilakukan.

Bab II Landasan Kepustakaan

Pada bab ini akan membahas tentang landasan teori atau tinjauan pustaka yang digunakan dalam mendukung jalannya penelitian ini. Tinjauan pustaka akan digunakan sebagai acuan perbandingan penelitian yang akan dilakukan dengan berbagai penelitian yang sebelumnya telah dilaksanakan oleh peneliti lain. Landasan teori akan berisi berbagai metode yang akan digunakan penulis untuk mencapai tujuan dalam pelaksanaan penelitian yang dilakukan.

Bab III Metodologi Penelitian

Pada bab ini akan membahas tentang tahapan yang dilakukan dalam melaksanakan penelitian ini. Pada bab ini juga akan dibahas metode apa saja yang akan digunakan untuk mendukung tercapainya sistem yang akan dibangun dan langkah-langkah dalam menggunakan metode tersebut.

Bab IV Analisis Kebutuhan

Pada bab ini akan membahas tentang fungsionalitas dari sistem serta pengguna yang dituju, sehingga sistem yang dibangun memiliki fungsi-fungsi yang valid dan benar-benar dibutuhkan oleh pengguna.

Bab V Perancangan

Pada bab ini akan membahas mengenai rancangan dari sistem yang digunakan sebagai dasar dalam membangun sistem yang akan diimplementasikan, sehingga akan mempermudah didalam melakukan implementasi serta proses implementasi tidak keluar dari ketentuan yang ditetapkan.

Bab VI Implementasi

Pada bab ini akan membahas tentang implementasi dari metode yang digunakan ke dalam sistem sehingga menyelesaikan permasalahan yang dibahas dalam penelitian ini yakni, penilaian esai secara otomatis. Pada bagian ini, langkah-langkah dalam menggunakan metode yang dipakai akan diterapkan dalam sistem.

Bab VII Pengujian

Pada bab ini akan membahas tentang hasil dari sistem yang telah dirancang. Bagian ini merupakan tahapan akhir dari sistem yang akan dibangun. Pada bab ini

juga akan dimuat hasil dari pengujian sistem apakah dapat menyelesaikan masalah yang telah dirumuskan dan seberapa besar tingkat akurasi dari sistem. Dari hasil pengujian tersebut akan didapatkan kesimpulan apakah sistem ini dapat digunakan atau tidak di masa yang akan datang.

Bab VIII Penutup

Pada bab ini akan dibahas tentang kesimpulan dan saran dari hasil penelitian yang telah dilakukan oleh penulis. Kesimpulan tersebut dapat berupa penjelasan bahwasannya metode yang digunakan dalam permasalahan ini sudah dapat diterapkan atau belum, serta saran dapat berupa tambahan yang dapat digunakan untuk meningkatkan hasil dari penelitian ini jika dilakukan penelitian lebih lanjut



BAB 2 LANDASAN KEPUSTAKAAN

1.1 Kajian Pustaka

Penelitian pertama dengan judul “*Sentiment Analysis for Bullying Word Detection in Social Network*” dilakukan oleh Srilakshmi. Dari penelitian tersebut digunakan salah satu metode NLP (*Natural Language Processing*), yaitu analisis sentimen dengan metode klasifikasi yang digunakan adalah *Naive Bayes Classifier*. Dari penelitian tersebut data disediakan dengan menginputkan text yang berasal dari media sosial. Hasil yang disimpulkan pada penelitian tersebut menyatakan bahwa analisis sentimen dapat digunakan untuk memprediksi opini pengguna media sosial, sehingga dapat mencegah kata-kata yang bersifat *bullying* di dalam media sosial.

Pada penelitian kedua membahas tentang metode mendeteksi cybercrime di dalam komunikasi daring dengan kasus yang diangkat yaitu *cyberbullying*. Penelitian ini dilakukan oleh Al-garadi, Varathan, dan Ravana, dengan penelitian yang berjudul “*Cybercrime detection in online communications: The experimental case of cyberbullying detection in the Twitter*”. Pada penelitian tersebut dilakukan perbandingan performa antara beberapa algoritme *machine learning* yang digunakan untuk mendeteksi *cyberbullying*, dengan data yang disediakan berupa pemilihan secara acak tweet yang telah diambil menggunakan API Twitter. Dari penelitian tersebut didapatkan hasil bahwa algoritme *machine learning*, seperti algoritme random forest, naive bayes dapat digunakan untuk mendeteksi *cyberbullying*.

1.2 Dashboard

Dashboard merupakan tampilan visual yang berisi informasi penting yang diperlukan didalam mencapai suatu tujuan yang dapat diatur didalam satu buah layar sehingga lebih mudah untuk dipantau pengguna (Januarita & Dirgahayu, 2015). *Dashboard* menampilkan beberapa informasi yang secara komponen bersatu dalam mencapai tujuannya. Pada umumnya *dashboard* memiliki satu halaman utama yang didalamnya memiliki beberapa informasi sesuai dengan tujuan *dashboard* itu sendiri. *Dashboard* memberikan tampilan antarmuka dengan berbagai bentuk seperti diagram, laporan, indikator visual, yang dipadukan dengan informasi dinamis yang relevan.

Sebuah *dashboard* yang dibangun dapat memberikan beberapa manfaat, yaitu sebuah *dashboard* dapat mengkomunikasikan sebuah tujuan kepada pihak-pihak yang berkepentingan dalam sebuah organisasi. *Dashboard* dapat digunakan untuk memonitor pelaksanaan rencana dan strategi, sehingga pihak-pihak yang berkepentingan dapat mengidentifikasi sebuah masalah dan membuat strategi untuk mengatasinya. *Dashboard* dapat menyampaikan wawasan dan informasi ke semua pihak, dengan sajian informasinya yang secara sekilas menggunakan grafik, simbol, bagan, dan warna-warna yang memudahkan pengguna untuk memahami dan mempersepsikan informasi secara benar (Hariyanti, 2008).

Dalam pembangunan sebuah *dashboard* terdapat beberapa prinsip-prinsip mendasar yang perlu diperhatikan dalam merancang pembangunan *dashboard* yaitu, menyajikan informasi dengan tujuan yang spesifik, mensinergikan informasi dari berbagai aspek dalam sebuah tampilan, responsif serta interaktif, memungkinkan untuk menganalisis kondisi sebelumnya, memonitor kondisi saat ini dan memprediksi tren kedepan (Hariyanti, 2008). Berdasarkan level manajemen yang didukungnya, *dashboard* dapat dikelompokkan menjadi *dashboard* level operasional, analitikal/taktikal, dan strategik. *Dashboard* level operasional memberikan informasi mengenai aktifitas yang terjadi, *dashboard* level analitikal memberikan informasi yang diperlukan oleh analis untuk mengetahui penyebab suatu kejadian, sedangkan *dashboard* level strategik memberikan informasi untuk membuat keputusan bisnis (Hariyanti, 2008).

1.3 Cyberbullying

Cyberbullying merupakan segala bentuk kekerasan yang dilakukan di dalam dunia maya. *Cyberbullying* dapat berbentuk ancaman, hinaan, ejekan, dan intimidasi dari pelakunya. *Cyberbullying* merupakan bentuk *bullying* yang terjadi ketika seseorang menggunakan teknologi informasi dan komunikasi seperti *email*, pesan teks, pesan singkat, *website* pribadi situs jejaring Facebook, Twitter, Tumblr, dan permainan daring, untuk digunakan secara sengaja, berulang-ulang dengan perilaku yang tidak ramah yang dimaksudkan untuk merugikan orang lain. Tindakan *cyberbullying* dapat menciptakan ketidak seimbangan kekuatan antara korban dan pelakunya (Besley, 2009). Munculnya tindakan *cyberbullying* ini disebabkan oleh perkembangan pesat teknologi informasi dan komunikasi serta penggunaan perangkat yang dapat terhubung internet oleh remaja dan anak sekolah (Cassidy, et al., 2013). Adanya peningkatan penggunaan teknologi informasi dan komunikasi melalui smartphone, tablet, dan laptop menciptakan adanya *digital divide* (Pearce & Rice, 2017) antara generasi muda dan generasi yang lebih tua, sehingga seringkali orang tua tidak menyadari bahaya teknologi untuk anak-anak mereka (Dehue, et al., 2008).

Orang tua sebagai agen pertama dalam kehidupan sosial seorang anak seringkali kesulitan untuk melakukan komunikasi secara langsung dengan anaknya dikarenakan adanya penggunaan teknologi yang berlebihan, sehingga dari sinilah muncul peluang seorang anak terkena *cyberbullying* tidak terdeteksi oleh orangtuanya. Salah satu pembeda utama kejahatan *cyberbullying* dengan kejahatan kriminal yakni pada anonimitas kejahatan, sehingga pada kasus *cyberbullying* memungkinkan pelaku untuk tidak dikenal atau diketahui. Bentuk-bentuk *cyberbullying* hampir menyerupai tindakan *bullying* yang dilakukan di dunia nyata hanya saja pada *cyberbullying* dapat dilakukan lebih luas karena dilakukan pada dunia maya. Pelaku *cyberbullying* memiliki karakteristik agresif dan intimidatif ketika menyerang korbannya. Bentuk agresif dan intimidatif pelaku *cyberbullying* dapat seperti, pelaku mengirimkan komentar *cyberbullying* berulang kali, mengancam kebebasan berinternet korban, mengancam keselamatan korban, mendominasi orang lain, serta tidak berempati terhadap perasaan orang lain.

Berdasarkan penelitian yang dilakukan oleh Prince dan Dalglish (2010), Jenis *cyberbullying* yang dilakukan pelaku *cyberbullying* dapat berbentuk seperti, pemberian nama negatif kepada korbannya, ini adalah bentuk serangan untuk memberikan label buruk kepada korban, nama negatif yang dimaksud adalah nama hewan, nama makhluk halus, dan panggilan fisik. Lalu bentuk lainnya dari *cyberbullying* yakni dapat seperti penyebaran foto, penyebaran foto ini dimaksudkan pelaku untuk membuat malu korbannya, serta untuk dijadikan objek hiburan kepada pelaku dan orang lain. bentuk lainnya yaitu, mengancam keselamatan fisik dari korbannya dan pendapat yang merendahkan (Price & Dalglish, 2010).

Di Indonesia kasus *cyberbullying* yang mendapat banyak perhatian media terjadi pada Sonya Depari. Saat itu Sonya Depari sebagai korban *cyberbullying* awalnya mengaku sebagai anak dari Kapolda Riau yaitu Arman Depari ketika korban sedang diperiksa oleh seorang polisi, namun diketahui bahwa pengakuan Sonya Depari tidak benar dan dia bukan anak dari Pejabat Riau tersebut. Setelah saat itu Sonya Depari mendapatkan bullying pada sosial media nya dari para pelaku. Banyak kalimat yang digunakan pelaku *cyberbullying* dalam melakukan perbuatannya kepada Sonya Depari, seperti dalam salah satu postingan di sosial media Sonya Depari terdapat pelaku *cyberbullying* yang mengatakan “penipu kelas kakap lo!!!! Cuih kelaut aje !”, “kasiannya yang Cuma bisa ngaku-ngaku doang, jangan-jangan kau anak jenderal alam mimpi.”. Tindakan *cyberbullying* bahkan pernah dirasakan oleh tokoh masyarakat, seperti yang pernah diterima Ahmad Dhani didalam media sosial twitter yang dimilikinya. Contoh tindakan *cyberbullying* dapat dilihat pada Gambar 2.1, Gambar 2.2, dan Gambar 2.3.



Gambar 0.1 Contoh Kasus *Cyberbullying* di Indonesia



Gambar 0.2 Contoh Kasus *Cyberbullying* di Indonesia Pada Media Sosial Twitter



Gambar 0.3 Contoh Kasus *Cyberbullying* di Indonesia Pada Media Sosial Twitter

1.3.1 Dampak *Cyberbullying*

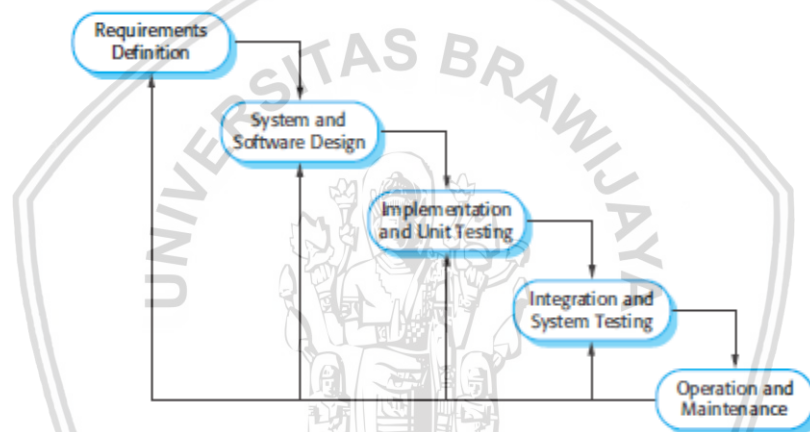
Cyberbullying dapat mempengaruhi kondisi mental dari korbannya. Hal ini juga berdampak pada kesehatan fisik secara keseluruhan. Menurut dinas kesehatan ada dua dampak utama yang ditimbulkan yakni secara fisik, dan secara mental. Secara fisik korban dapat mengalami, hilangnya selera makan, gangguan tidur, gangguan pencernaan, hingga jantung berdebar-debar. Secara mental korban dapat mengalami gejala seperti, gelisah, depresi, rasa harga diri berkurang, murung, gangguan emosional, hingga bunuh diri. Investigasi terhadap dampak yang ditimbulkan dari *cyberbullying* memperlihatkan bahwa konsekuensi psikologis dan emosional dari *cyberbullying* merupakan masalah terbesar bagi korban (Dredge, et al., 2014).

Dalam beberapa kasus yang ditemukan kepada korban kasus *cyberbullying*, mengakibatkan korban menjadi malas untuk sekolah, melakukan pembolosan, prestasi akademik menurun dan tidak merasa aman di sekolah (Cassidy, et al., 2013). Hal ini tentunya dapat mengintimidasi para korban sehingga korban enggan untuk melakukan kontak sosial, padahal dalam rentang usia remaja,

perkembangan psikis merupakan unsur penting didalam hidup, karena perkembangan yang terjadi pada manusia terjadi hingga manusia tersebut meninggal, sehingga ketika seseorang mengalami gangguan psikis, itu akan terus berdampak dari waktu ke waktu selama seseorang tersebut berkembang sepanjang hidupnya.

1.4 Model Pengembangan *Waterfall*

Model pengembangan *waterfall* merupakan salah satu model proses pengembangan perangkat lunak. Menurut Ian Sommerville (2011), proses pengembangan perangkat lunak adalah sekumpulan aktivitas yang berkaitan yang bertujuan untuk menghasilkan sebuah produk perangkat lunak, sedangkan model proses pengembangan perangkat lunak adalah representasi dari proses pengembangan perangkat lunak. Tahapan model proses pengembangan perangkat lunak waterfall dapat dilihat pada Gambar 2.4.



Gambar 0.4 Model Pengembangan *Waterfall*

Sumber : Sommerville (2011)

Menurut Sommerville (2011) terdapat beberapa tahapan utama didalam model *waterfall* yang menggambarkan dasar dari aktivitas pengembangan perangkat lunak yaitu :

1. Analisis Kebutuhan

Proses menetapkan layanan, batasan, dan tujuan dari sistem yang dibangun dengan berkonsultasi dengan pengguna. Hasil dari kebutuhan yang didefinisikan pengguna menjadi dasar dari spesifikasi sistem.

2. Desain Perangkat Lunak

Menetapkan keseluruhan arsitektur sistem berdasarkan kebutuhan yang telah ditetapkan sebelumnya. Proses desain perangkat lunak melibatkan identifikasi dan mendeskripsikan dasar abstraksi perangkat lunak dan hubungan didalam sistem.

3. Implementasi dan Pengujian Unit

Pada tahap ini, desain perangkat lunak di realisasikan kedalam kode program atau kumpulan unit program. Pengujian unit melibatkan verifikasi terhadap setiap unit didalam spesifikasi kebutuhan.

4. Integrasi dan Pengujian sistem

Setiap unit dari program yang terintegrasi di uji sebagai sistem yang lengkap untuk memastikan perangkat lunak telah memenuhi kebutuhan. Setelah melalui pengujian, sistem diserahkan kepada pengguna.

5. Operasi dan Pemeliharaan

Pada umumnya ini merupakan fase daur hidup terlama, di tahap ini sistem di pasang untuk digunakan oleh pengguna. Pemeliharaan melibatkan perbaikan kesalahan yang tidak ditemukan pada tahap pengujian sebelumnya, meningkatkan implementasi dari unit sistem dan menambah layanan sistem jika terdapat kebutuhan baru yang ditemukan.

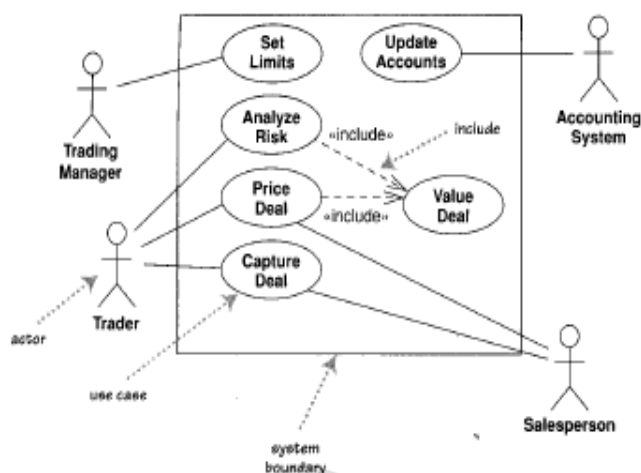
1.5 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi dalam industri untuk melakukan visualisasi, merancang, dan mendokumentasikan sistem perangkat lunak (Dharwiyanti dan Wahono, 2003). UML mendefinisikan berbagai macam notasi. Notasi UML merupakan sebuah bentuk khusus untuk menggambarkan berbagai macam diagram di dalam perangkat lunak. Pengembangan UML dikoordinasikan oleh Object Management Group (OMG), dan hingga saat ini UML telah menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek. UML memiliki berbagai macam diagram, menurut kebutuhannya, namun pada umumnya diagram yang sering digunakan yaitu :

1. Use Case

Use case merupakan sebuah teknik untuk menangkap kebutuhan fungsional dari sebuah sistem. *Use case* bekerja dengan mendeskripsikan interaksi antara pengguna dengan sistem, dan sistemnya sendiri dengan menyediakan menceritakan bagaimana sistem digunakan (Fowler, 2004).

Use case diagram menunjukkan *aktor*, *use case*, dan relasi antara keduanya, dimana *aktor* adalah sebuah peran yang pengguna lakukan didalam sebuah sistem, sedangkan *use case* adalah sekumpulan skenario yang secara bersama menggambarkan tujuan dari pengguna didalam sebuah sistem. Contoh sebuah *use case* diagram dapat dilihat pada Gambar 2.5 berikut :



Gambar 0.5 Use Case Diagram

Sumber : Fowler (2004)

2. Sequence diagram

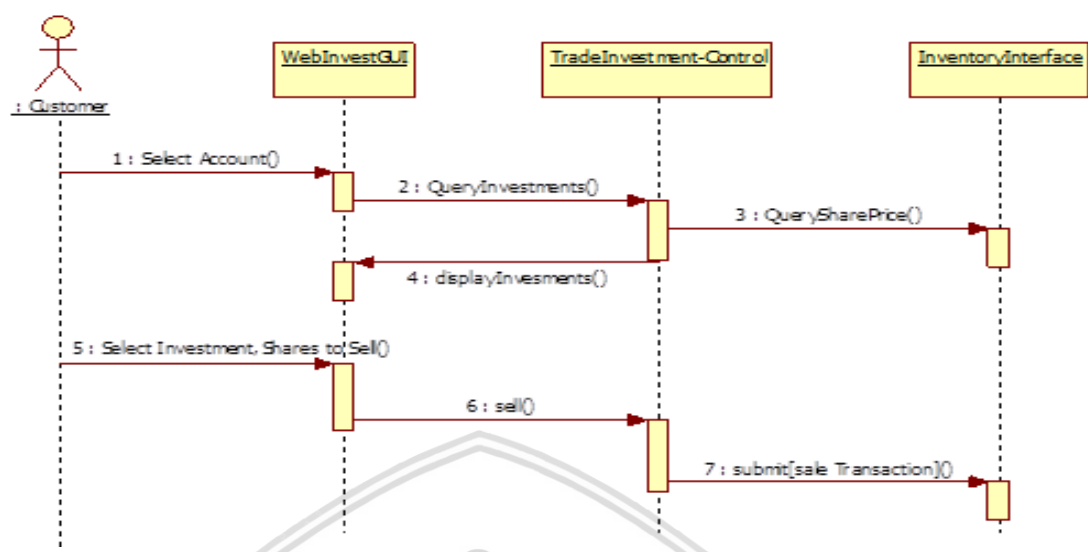
Sequence diagram menggambarkan perilaku dari sebuah skenario, *sequence diagram* juga menunjukkan sejumlah objek dan pesan yang saling berinteraksi diantara objek tersebut (Fowler, 2004). *Sequence diagram* digunakan untuk menggambarkan skenario atau langkah-langkah yang dilakukan sebagai respon dari sebuah event untuk menghasilkan output tertentu. *Sequence diagram* memiliki komponen-komponen seperti yang terdapat pada Tabel 2.1.

Tabel 0.1 Tabel Komponen Sequence Diagram

| Komponen | Keterangan |
|----------|-------------------|
| | Aktor |
| | Object |
| | Combined Fragment |
| | Frame |
| | Stimulus |
| | Self Stimulus |

Sumber : Arief (2012)

Contoh *Sequence diagram* dapat dilihat pada Gambar 2.6 berikut :






Gambar 0.6 Contoh *Sequence Diagram*
Sumber : Arief (2012)

3. *Class Diagram*

Class diagram menggambarkan tipe-tipe dari objek didalam sistem, dan berbagai jenis variasi dari relasi yang ada diantara objek-objek tersebut (Fowler, 2004). *Class diagram* juga menunjukkan properti dan operasi yang ada didalam sebuah *class*, serta batasan dari objek yang berhubungan. *Class diagram* memiliki komponen yang digambarkan dalam Tabel 2.2 berikut :

Tabel 0.2 Tabel Komponen *Class Diagram*

| Komponen | Nama komponen | Keterangan |
|---------------------------------------------------------------------------------------------------------------------------------------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <div><div>Nama <i>Class</i></div><div>+ atribut + atribut + atribut</div><div>+ <i>method</i> + <i>method</i></div></div> | <i>Class</i> | Sebuah <i>class</i> digambarkan dengan sebuah kotak dengan tiga bagian, bagian atas adalah nama dari <i>class</i> , bagian tengah mendefinisikan properti/ atribut <i>class</i> , dan bagian akhir mendefinisikan method dari sebuah <i>class</i> . |
| <div><div>1..n</div><div>Owned by</div><div>1</div></div> | <i>Association</i> | Sebuah asosiasi merupakan hubungan yang umum antara 2 class dan dilambangkan oleh sebuah garis yang menghubungkan 2 <i>class</i> . Pada asosiasi juga menampilkan hukum |

| Komponen | Nama komponen | Keterangan |
|-----------------------------------------------------------------------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | | multiplisitas. (Contoh : <i>one-to-one, one-to-many, many-to-many</i>). |
|  | <i>Composition</i> | Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan merupakan bagian dari <i>class</i> lain maka <i>class</i> tersebut memiliki relasi <i>composition</i> terhadap <i>class</i> tempat dia bergantung. |
|  | <i>Dependency</i> | Umumnya penggunaan hubungan <i>dependency</i> digunakan untuk menunjukkan operasi suatu <i>class</i> yang menggunakan <i>class</i> lain. |
|  | <i>Aggregation</i> | <i>Aggregation</i> memiliki hubungan seperti <i>composition</i> , namun pada agregasi hubungannya tidak sekuat <i>composition</i> , dan kelas yang bergantung masih tetap dapat berdiri sendiri jika <i>class</i> tempat dia bergantung tidak ada. |

Sumber : Booch (1999)

1.6 Twitter

Twitter merupakan platform *micro-blogging* yang dapat digunakan setiap orang untuk membagikan informasi atau yang biasa disebut *tweet* dengan maksimum karakter sebanyak 140 karakter (Makice, 2009). Dengan karakteristiknya yang menyerupai *micro-blogging* tersebut, sosial media ini pun menjadi salah satu tempat favorit yang digunakan pengguna sebagai sarana informasi, edukasi, maupun sarana berbagi perasaan setiap penggunanya. Fungsi-fungsi yang disediakan Twitter ini juga beragam, seperti fungsi *tweet*, untuk menuliskan informasi, fungsi *retweet* untuk membagikan informasi, fungsi *reply* untuk membalas, fungsi *follow* untuk mengikuti informasi dari pengguna lainnya. Hingga saat ini Twitter pun masih menjadi salah satu sosial media terbesar yang memiliki banyak pengguna, bahkan beberapa orang yang berpengaruh di Indonesia memiliki akun Twitter seperti bapak presiden Republik Indonesia Joko Widodo yang menggunakan Twitter untuk melakukan komunikasi langsung dengan seluruh pengguna Twitter khususnya pengguna di Indonesia.

Sejak didirikan pada tahun 2006 media sosial yang didirikan oleh Jack Dorsey dan memiliki logo burung ini memiliki pengguna aktif Twitter telah berkembang sangat pesat, hingga tahun 2014 tercatat terdapat 284 juta pengguna aktif Twitter diseluruh dunia yang mengirimkan sekitar 500 juta *tweet* perharinya (Carley, et al., 2015). Di Indonesia sebuah firma market research mencatat pengguna Twitter pada tahun 2012 sebanyak 29.4 juta pengguna atau 11 persen dari seluruh jumlah penduduk di Indonesia, dan menjadikan negara Indonesia sebagai negara peringkat ke 5 dengan penggunaan Twitter terbanyak di dunia. Pada penelitian lainnya menemukan bahwa Indonesia memiliki *tweet*/pengguna dengan rasio terbesar yaitu 1.813,53 diatas Jepang (1.617,35) dan Brazil (1.370,27) (Poblete, et al., 2011)

Penggunaan media sosial Twitter dalam perjalanannya hingga saat ini tidak hanya sebatas sebagai media komunikasi bagi penggunanya, banyak pengguna-pengguna yang sangat kreatif yang memanfaatkan Twitter, seperti Twitter yang digunakan untuk media berita. Banyak akun media berita resmi yang memanfaatkan Twitter sebagai penyalur berita yang dikelolanya seperti CNN Indonesia, BBC Indonesia, detik, dan liputan 6. Twitter juga dapat digunakan sebagai sarana pembelajaran ilmu pengetahuan, seperti akun yang dikelola NASA yang terdapat 28 juta pengguna yang membagikan informasi mengenai pengetahuan tentang peristiwa atau kejadian alam. Twitter juga dapat digunakan sebagai sarana informasi keadaan darurat dan genting, seperti terjadinya bencana alam yang pada umumnya diinformasikan oleh akun resmi BMKG (Makice, 2009).

1.6.1 Twitter API

Perusahaan Twitter menyediakan Twitter API yang dapat digunakan oleh pengembang atau peneliti untuk dapat mengembangkan aplikasi dengan menggunakan data yang tersedia dari Twitter. Pengguna dapat melakukan pendaftaran pada Twitter *developer* untuk mendapatkan key sebagai penanda pengguna yang menggunakan API tersebut. Pada API Twitter tersedia fungsi yang dapat dijalankan oleh pengguna, seperti fungsi untuk mengatur akun pengguna api tersebut, selain itu terdapat fungsi untuk melakukan filter *tweet* secara *realtime*, fungsi untuk melakukan direct message, serta Twitter juga menyediakan fungsi untuk mengakses data *tweet* yang tersimpan di dalam database Twitter, sehingga pengguna dapat melakukan ekstraksi data *tweet* yang tersimpan. Pada fungsi untuk mengakses data *tweet*, setiap datanya mengandung informasi seperti *user id*, *username*, *user biography*, *user screenname*, isi dari *tweet*, waktu dari *tweet* dibuat, dll (Al-garadi, et al., 2016).

Pada umumnya penggunaan Twitter API seringkali digunakan untuk pengambilan data dalam sebuah penelitian, seperti penelitian yang dilakukan oleh Al-garadi, et al., yang mengenai deteksi *cybercrime* didalam komunikasi online, dengan studi kasus deteksi *cyberbullying* di jaringan Twitter, didalam penelitian tersebut pada bagian metodologi digunakan API Twitter dengan fungsi mengakses data *tweet* didalam pengambilan datanya. Pada API Twitter menyediakan beberapa referensi penggunaan pencarian data didalam API nya, yaitu standar

search API, enterprise search API, dan premium search API. Dalam standar search API Twitter menawarkan pencarian data dari tweet berdasarkan *query* yang ditentukan. Pada enterprise search API Twitter menawarkan pengambilan data yang lebih besar pada skala bisnis dengan waktu yang disediakan yaitu pencarian *tweet* dengan *tweet* yang di posting dalam 30 hari sebelumnya, dan *tweet* dari awal Twitter diciptakan pada maret tahun 2006. Pada premium search API pada umumnya hampir sama dengan yang ditawarkan enterprise search API hanya terdapat perbedaan pasar yang di tuju oleh Twitter.

1.7 Google Maps

Google Maps merupakan salah satu produk yang dihasilkan oleh Google yang diluncurkan pada tahun 2005 berupa layanan navigasi dan pemetaan yang dapat diakses melalui web. Hingga saat ini Google Maps menyediakan layanan pencitraan melalui satelit, kondisi real-time lalu lintas, serta tampilan jalanan di suatu tempat. Google Maps diciptakan dan didesain pertama kali oleh Lars dan Jens Rasmussen. Hingga saat ini Google Maps dapat digunakan pada platform *web* maupun *mobile*. Berdasarkan statistik yang dirilis oleh Google, hingga saat ini penggunaan Google Maps telah mencapai 250 juta mil atau sekitar 402,3 km setiap hari penggunaannya atau dapat dikatakan sebagai layanan navigasi dan pemetaan terbesar yang banyak digunakan diseluruh dunia.

Layanan Google Maps memanfaatkan teknik *geolocation* untuk menemukan lokasi yang diinginkan pengguna. Pada penggunaan *geolocation* ada beberapa metode yang digunakan untuk menemukan lokasi, yaitu dengan IP *address*, sembugan *wireless*, BTS, dan *embeded* GPS pada telepon seluler. Dalam teknik pencariannya *geolocation* memerlukan data koordinat latitide dan longitude untuk menuju lokasi pengguna (Mahdia & Noviyanto, 2013). Layanan Google Maps merupakan layanan yang interaktif, didalamnya pengguna dapat menggeser lokasi, mengubah level *zoom*, serta mengubah tampilan jenis peta sesuai dengan keinginan pengguna. Google Maps mempunyai sistem koordinat yang sama dengan google earth yaitu world geodetic system 1984 (WGS-84) dan memiliki proyeksi peta Mercator (Kusuma, & Budisusanto, 2015).

1.7.1 Google Maps API

Google Maps API merupakan salah satu teknologi yang dikembangkan oleh Google yang dapat diberdayakan oleh peneliti untuk menciptakan penemuan baru dengan menyertakan Google Maps didalam pengembangannya. Layanan ini merupakan aplikasi antarmuka yang dapat diakses melalui javascript untuk mengintegrasikan Google Maps kedalam aplikasi yang dibangun. Versi API yang terakhir diluncurkan oleh Google diluncurkan pada tahun 2009 pada saat konferensi Google I/O . API pada Google Maps terdiri dari file javascript yang berisi kelas dengan metode beserta propertinya, serta dalam penggunaannya google maps ini mengombinasikan antara HTML, CSS, dan Javascript (Svennerberg, 2010).

Pengguna harus melakukan pendaftaran untuk mendapatkan api key sebelum dapat memanfaatkan layanan yang disediakan Google Maps. API key merupakan

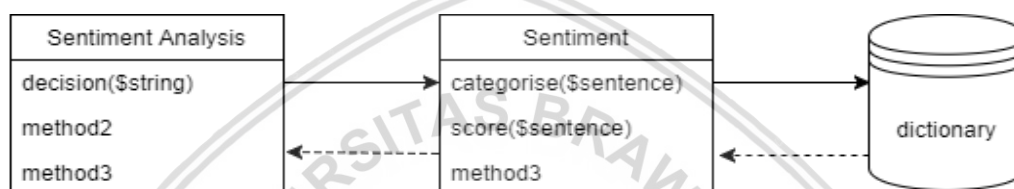
kode unik yang digenerasikan oleh Google ketika pengguna melakukan pendaftaran, yang digunakan server google untuk mengenali client nya. Google Maps juga tidak memberikan biaya kepada penggunanya untuk melakukan pendaftaran. Pada umumnya layanan Google Maps API digunakan dalam membangun aplikasi berbasis GIS atau sistem informasi geografis. API yang disediakan oleh google terdapat beberapa jenis, yaitu API untuk Google Maps android, Google places API untuk android, Google Maps SDK untuk iOS, Google Places API untuk iOS, dan yang umumnya digunakan untuk membangun aplikasi berbasis web yaitu Google Maps javascript API.

1.8 Analisis Sentimen

Analisis sentimen atau yang biasa disebut *option mining* merupakan proses menyajikan informasi dengan membangun sebuah sistem yang dapat mengklasifikasikan dokumen teks kedalam dua kategori yakni nilai negatif dan positif sesuai dengan keseluruhan sentimen yang dinyatakan ke dalam dokumen tersebut (Hidayat, 2015). Penelitian mengenai analisis sentimen mulai sering dilakukan pada tahun 2002, salah satu contohnya yaitu penelitian yang dilakukan oleh Turney mengenai *review* konsumen atas suatu produk. Pada analisis sentimen terdapat kombinasi antara text mining dan natural language processing, dimana pada bagian text miningnya terdapat sebuah metode/perhitungan yang dilakukan untuk menyelesaikan kasusnya. Salah satu metode yang dapat menyelesaikan masalah analisis sentimen yaitu naive bayes classifier (Rozi, et al., 2012). Sejalan dengan perkembangan ilmu pengetahuan saat ini terdapat banyak terdapat *library* yang menyediakan fungsi untuk analisis sentimen, salah satunya yakni yang dikembangkan oleh James Hennessey yang di simpan melalui repository github dengan nama *phpinsight*. Pada *library* yang dikembangkan James tersebut dapat digunakan untuk pengembangan aplikasi web dengan didalamnya terdapat fungsi untuk melakukan klasifikasi sentimen dengan menggunakan kata-kata yang telah tersimpan dan dikategorikan sebagai kata positif, negatif, maupun netral menggunakan algoritme naive bayes dalam melakukan kalkulasinya untuk menentukan klasifikasi dari sentimen yang dianalisis.

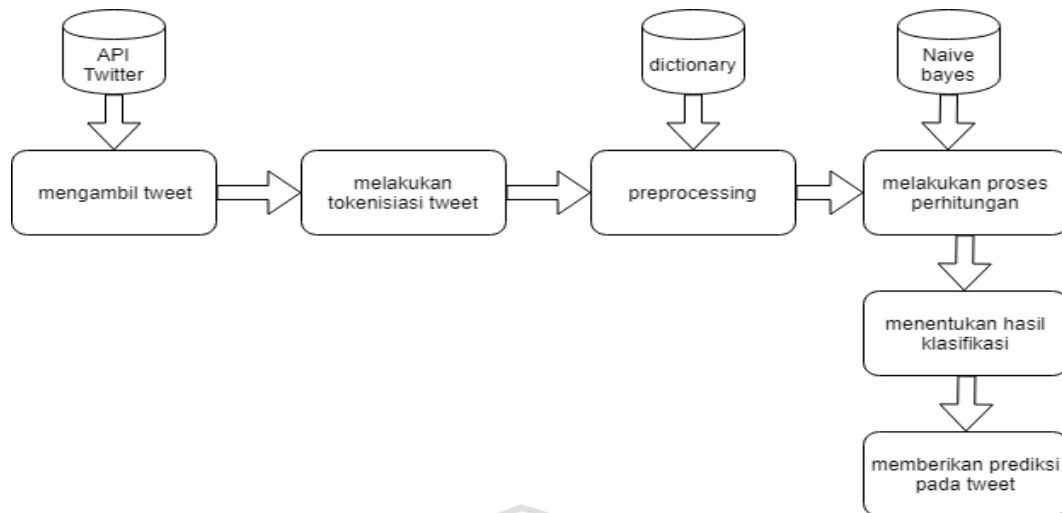
Dalam penelitian ini terdapat *library* yang digunakan untuk melakukan analisis sentimen, dengan *library* yang bernama *laravel-sentiment-analysis* yang diciptakan oleh Antoine Augusti dan dibagikan dalam repository github. Dalam *library* yang digunakan terdapat sebuah kamus kata yang menyimpan kata-kata positif negatif maupun netral yang digunakan dalam perhitungan algoritma klasifikasinya. *Library* yang diciptakan Antoine Augusti hanya memiliki kamus yang memuat kata-kata dalam bahasa inggris. Untuk keperluan penelitian ini terdapat perubahan kamus yang memuat kata-kata pendapat negatif maupun positif dalam bahasa indonesia yang didapatkan dari repository github yang dirilis oleh David Haryalesmana berdasarkan penelitian Wahid, D. H., & Azhari, S. N. (2016), serta terjemahan dari kata-kata yang didapatkan dalam penelitian liu, Bing & Cheng Junsheng (2005) yang berjudul "Opinion Observer: Analyzing and Comparing Opinions on the Web".

Pada *library* yang digunakan dalam penelitian ini memiliki beberapa kelas yang dapat diakses fungsinya, dengan fungsi utamanya yaitu melakukan kategorisasi terhadap suatu kalimat. Dalam tahapan kerja *library* ini controller sebuah *framework* yang ingin menggunakan fungsi analisis sentimen harus memanggil fungsi `decision(string)`, setelah itu kelas Sentiment Analysis memanggil fungsi `categorise(sentence)` pada kelas sentiment dan memanggil kelas `score(sentence)`. Fungsi `score(sentence)` terdapat pengecekan setiap kata pada sebuah kalimat untuk dicocokkan dengan *dictionary* yang ada dan pada fungsi `score(sentence)` terdapat perhitungan dari setiap kategori, yaitu positif, negatif, dan netral. Kemudian pada kelas Sentiment akan mengembalikan salah satu kategori dengan nilai perhitungan tertinggi dan akan dibalikkan ke kelas yang memanggilnya yaitu kelas Sentiment Analysis. Ilustrasi tahapan kerja *library* sentimen analisis yang digunakan dalam penelitian ini dapat dilihat pada Gambar 2.7 berikut.



Gambar 0.7 Alur Kerja *Library* Sentimen Analisis

Analisis sentiment di dalam penelitian ini berintegrasi dengan data yang didapatkan dari API Twitter. Sebelum *Library* analisis sentiment melakukan klasifikasi, data berupa *tweet* yang didapatkan dari API Twitter, dikirimkan ke dalam sebuah fungsi di kelas *library* analisis sentimen. Kemudian dari setiap kalimat di dalam *tweet* yang dikirim ke dalam fungsi tersebut dilakukan *tokenisasi* atau dipecah menjadi perkata untuk dicek kedalam *dictionary* yang berisi daftar kata-kata positif, negatif, maupun netral yang terdapat di dalam *library* untuk dilakukan perhitungan. Dari proses perhitungan tersebut kemudian *library* akan memberikan label pada sebuah kalimat yang sebelumnya telah diproses untuk dikategorikan ke dalam kalimat positif, negatif, maupun netral.

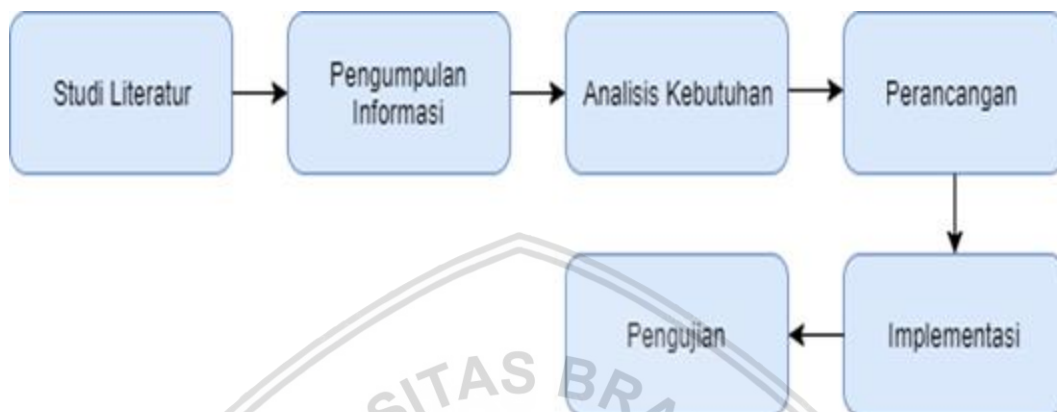


Gambar 0.8 Alur Integrasi Library Analisis Sentimen dan API Twitter



BAB 3 METODOLOGI

Pada bagian ini menjelaskan langkah-langkah yang akan dilakukan didalam Pengembangan *dashboard* sistem persebaran kasus *cyberbullying* pada media sosial Twitter. Secara umum pelaksanaan pembuatan perangkat lunak dilakukan dengan mengikuti alur kerja yang dapat dilihat pada Gambar 3.1 berikut.



Gambar 0.1 Diagram Alur Kerja

1.1 Studi Literatur

Studi literatur merupakan langkah untuk menjelaskan dasar teori yang digunakan dalam penelitian yang berfungsi untuk menunjang penulisan didalam penelitian. Literatur dapat berupa buku, teks, paper, dan literatur daring. Studi literatur dilakukan dengan mengumpulkan dan mempelajari literatur yang berkaitan dengan Pengembangan *dashboard* sistem persebaran kasus *cyberbullying* pada media sosial Twitter. Teori-teori pendukung yang digunakan dalam pembangunan aplikasi ini, antara lain :

1. *Dashboard*
2. *Cyberbullying*
3. *Application Programming Interface* pada *Twitter*
4. *Library* analisis sentimen
5. *Application Programming Interface* pada *google maps*
6. Pengujian Unit, Pengujian Integrasi, dan Pengujian Validasi

Model pengembangan sistem yang digunakan didalam penelitian ini yaitu adalah model *waterfall*. Model *waterfall* cocok digunakan dalam mengembangkan sistem ini karena didasarkan kebutuhan sistem yang dibangun tidak memiliki kebutuhan yang berubah ubah.

1.2 Pengumpulan Informasi

Pengumpulan informasi pada penelitian ini dilakukan dengan mencari informasi terkait hal-hal yang nantinya digunakan didalam pengembangan *dashboard* sistem persebaran kasus *cyberbullying* pada media sosial Twitter. Informasi yang dibutuhkan didalam pengembangan sistem ini yaitu, penggunaan *library* pada API Twitter, penggunaan *library* analisis sentimen, serta penggunaan API google maps. *Library* pada API Twitter digunakan untuk mengakses fungsi-fungsi yang disediakan oleh Twitter yang didalam penelitian ini dikhususkan sebagai pengambilan data. Pada *library* analisis sentimen digunakan sebagai metode untuk mengklasifikasikan data yang sebelumnya didapatkan dari API Twitter. API google maps didalam penelitian ini digunakan untuk menampilkan lokasi dari data yang telah didapatkan dan sebagai menu utama dari *dashboard* sistem yang dibangun. Pengumpulan data pada penelitian ini menggunakan metode studi dokumen yang berasal dari layanan daring yang bersifat primer maupun sekunder.

1.3 Analisis Kebutuhan

Pada Analisis kebutuhan, dilakukan penetapan fungsi-fungsi yang diperlukan di dalam sistem yang akan dibangun, setelah data-data terkumpul. Di bagian ini semua fungsionalitas sistem dideskripsikan sehingga nantinya aplikasi yang dibangun sesuai dengan tujuan aplikasi, serta memenuhi semua kebutuhan dari penggunaannya. Kebutuhan yang telah didefinisikan nanti akan dimodelkan ke dalam pemodelan kebutuhan yang dinotasikan ke dalam bentuk *use case diagram* dimana *use case diagram* merupakan sebuah diagram yang menggambarkan fungsionalitas aplikasi dari perspektif pengguna, serta dijelaskan dengan *use case* skenario untuk menjelaskan detail dari fungsionalitas aplikasi ketika digunakan oleh pengguna.

1.4 Perancangan

Perancangan dilakukan setelah pemodelan kebutuhan didapatkan pada tahap analisis kebutuhan. Perancangan merupakan tahapan untuk mentransformasikan pemodelan kebutuhan ke dalam model perancangan, sehingga nantinya model perancangan dapat digunakan sebagai dasar untuk melakukan tahapan selanjutnya. Pemodelan perancangan pada penelitian ini dilakukan dengan memodelkan *sequence diagram*, *class diagram*, kemudian melakukan perancangan komponen, melakukan pemodelan basis data, serta antar muka. *Sequence diagram* merupakan penggambaran dari interaksi-interaksi pesan yang di kirim di dalam aplikasi. *Class diagram* merupakan penggambaran dari objek-objek yang berhubungan yang terdapat didalam sistem. Perancangan komponen digunakan untuk memberikan gambaran fungsi utama dari sistem dengan sebuah algoritme. Perancangan basis data dilakukan untuk mengidentifikasi entitas serta hubungan yang mendefinisikan ruang lingkup masalah yang ditangani oleh sistem. Perancangan antar muka dilakukan dengan memberikan gambaran dasar terhadap halaman yang menyusun sistem.

1.5 Implementasi

Implementasi merupakan tahapan yang mengacu kepada perancangan aplikasi. Di tahap ini dilakukan pembangunan sistem berdasarkan pemodelan yang dilakukan di tahap perancangan. Proses implementasi untuk aplikasi ini dilakukan dengan menggunakan Bahasa pemrograman PHP, HTML, Javascript, SQL dan menggunakan *framework* laravel. Pada proses implementasi terdapat tahapan untuk mengintegrasikan *library* yang digunakan yaitu, Twitter API, analisis sentimen, dan google maps API sebagaimana yang telah dipelajari pada tahapan pengumpulan informasi. Tahapan implementasi pada penelitian ini dilakukan dengan mendefinisikan spesifikasi perangkat keras, spesifikasi perangkat lunak, batasan implementasi. Setelah itu menuliskan implementasi beberapa kode program yang menjadi fungsi utama di dalam sistem yang dibangun, menotasikan implementasi basis data kedalam bentuk *physical data model*, serta menunjukkan implementasi antarmuka.

1.6 Pengujian

Pengujian dilakukan agar dapat menunjukkan bahwa aplikasi telah memenuhi kebutuhan dan mampu berjalan sesuai tujuan serta agar aplikasi yang dibangun bebas dari kegagalan. Pengujian dapat dilakukan pada saat aplikasi dibangun maupun saat selesai dibangun. Nantinya pengujian dilakukan dengan menggunakan metode pengujian *white box*, dan metode pengujian *black box*. Pada pengujian *white box* diwakili dengan pengujian unit, dan pengujian integrasi. Pada pengujian *black box* diwakili dengan pengujian validasi.

1.6.1 Pengujian Unit

Pengujian unit dilakukan untuk memastikan setiap unit terbebas dari kegagalan serta dapat berjalan sesuai tujuan yang diharapkan. Pengujian unit juga dapat dikatakan sebagai pengujian yang memperhatikan kode yang terdapat di dalam program yang dibangun. Pengujian unit pada penelitian ini menggunakan teknik *basis path testing*.

1.6.2 Pengujian Integrasi

Pengujian integrasi dilakukan dengan menguji interaksi antar class atau modul yang saling berhubungan. Pengujian integrasi yang dilakukan pada penelitian ini juga dilakukan dengan menggunakan basis path testing namun pada pengujian ini diawali dengan menganalisis proses yang terdapat interaksi antar kelas, kemudian memilih fungsi yang memiliki interaksi antar beberapa kelas tersebut.

1.6.3 Pengujian Validasi

Pengujian validasi dilakukan untuk memastikan seluruh kebutuhan yang telah didefinisikan telah diimplementasi sesuai dengan yang diharapkan. Pada pengujian ini, tahapan yang dilakukan yaitu membuat kasus uji untuk setiap kebutuhan fungsional, setelah itu dilakukan eksekusi setiap kasus uji dan membandingkan antara hasil yang diharapkan dengan hasil setelah di eksekusi.



BAB 4 ANALISIS KEBUTUHAN

Analisis kebutuhan merupakan tahapan awal yang dilakukan didalam membangun sebuah sistem. Pada tahapan ini dilakukan penentuan kebutuhan-kebutuhan yang diperlukan untuk membangun *dashboard* sistem untuk memonitoring kasus *cyberbullying* di media sosial Twitter, sehingga nantinya kebutuhan ini dapat menjadi dasar untuk membentuk fungsi-fungsi yang ada di dalam sistem yang dibangun. Pada analisis kebutuhan juga memperlihatkan siapa saja yang dapat berinteraksi dengan sistem.

1.1 Gambaran Umum Sistem

Berdasarkan permasalahan yang telah di gambarkan pada latar belakang, sistem ini dapat digunakan oleh seluruh stakeholder didalam permasalahan *cyberbullying* khususnya pemerintah melalui kementerian komunikasi dan informatika sebagai dasar dalam menggerakkan tindakan preventif. Didalam sistem ini pengguna dapat memilih lokasi untuk dipantau aktivitas *cyberbullying* nya berdasarkan kota yang tersedia, serta dapat memilih kota tertentu secara spesifik. Aktivitas *cyberbullying* yang dipantau yakni diambil dari layanan jejaring sosial Twitter, dengan *tweet* yang berada di sekitar lokasi yang sebelumnya dipilih oleh pengguna. *Tweet* yang masuk didalam pencarian lalu di filter menggunakan analisis sentimen untuk di kategorikan berdasarkan *tweet* yang mengandung unsur ujaran negatif dan yang tidak mengandung unsur ujaran negatif. *Tweet* yang mengandung unsur ujaran negatif akan ditampilkan presentasinya pada setiap kota yang tersedia, serta menampilkan diagram batang yang menampilkan jumlah detail dari *tweet* yang mengandung unsur ujaran negatif pada rentang waktu tertentu. Pengguna juga dapat melakukan pencarian data pada kota yang diinginkan oleh pengguna, serta dapat melihat bagaimana persentase pada kota tersebut dan detailnya.

1.2 Identifikasi Aktor

Pada identifikasi aktor ini digunakan untuk mengetahui siapa saja yang berhubungan didalam sistem ini. Berdasarkan gambaran umum sistem yang dibangun, aktor yang terdapat didalam sistem ini hanya aktor tunggal yaitu seorang pengguna, dimana pengguna ini dapat terdiri dari stakeholder yang nantinya dapat menggunakannya seperti pemerintah melalui kementerian komunikasi dan informatika. Deskripsi lebih detail mengenai peran aktor dapat dilihat pada Tabel 4.1.

Tabel 0.1 Tabel Identifikasi Aktor

| Aktor | Deskripsi |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Pengguna | Pengguna merupakan aktor yang dapat menjalankan fungsionalitas didalam sistem. Pengguna pada <i>dashboard</i> sistem monitoring <i>cyberbullying</i> ini dapat melihat data mengenai kasus <i>cyberbullying</i> yang ada di media sosial Twitter berdasarkan lokasi yang di inginkan dan ditampilkan dengan menggunakan Google Maps. |

1.3 Daftar Kebutuhan Fungsional

Kebutuhan fungsional merupakan suatu layanan yang harus disediakan didalam sebuah sistem berdasarkan tujuan awal dari sistem yang dibangun. Kebutuhan fungsional didalam sistem *dashboard* monitoring *cyberbullying* ini didapatkan dengan memahami domain dari permasalahan serta memahami bagaimana nantinya sistem ini dapat membantu pengguna didalam menyelesaikan masalahnya dengan melakukan elisitasi yaitu dengan metode wawancara kepada orang yang dekat dengan domain permasalahan. Setiap kebutuhan akan diberikan kode DMCF_XXYY, XX menunjukkan kebutuhan fungsional yang utama, dan YY menunjukkan spesifikasi kebutuhan fungsional yang utama. Daftar kebutuhan fungsional dan spesifikasinya dapat dilihat pada Tabel 4.2.

| No | Kebutuhan Sistem | Kode | Nama Use Case |
|-----|---------------------------------------------------------------------------------------------------------------------------------------------|-----------|-----------------------------------------|
| 1 | Sistem harus bisa mengambil data ujaran negatif pada interval waktu tertentu berdasarkan lokasi yang dipilih pengguna pada maps | DMCF_0100 | Mengambil data |
| 1.1 | Sistem menunjukkan lokasi yang dipilih pengguna dengan tanda berupa marker pada maps | DMCF_0101 | |
| 2 | Sistem harus bisa mengambil data ujaran negatif pada interval waktu tertentu berdasarkan lokasi yang diinputkan pengguna | DMCF_0200 | Mengambil data dengan form pencarian |
| 2.1 | Sistem harus menyediakan fasilitas untuk menginputkan lokasi berupa form pencarian | DMCF_0201 | |
| 3 | Sistem harus bisa mengambil data ujaran negatif pada interval waktu tertentu berdasarkan kota pilihan yang telah tersimpan didalam database | DMCF_0300 | Mengambil data pada kota yang tersimpan |
| 3.1 | Sistem harus menyediakan fasilitas berupa form yang berisi menu | DMCF_0301 | |

| No | Kebutuhan Sistem | Kode | Nama Use Case |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|----------------------------------------|
| | <i>dropdown</i> pilihan kota yang telah tersimpan sebelumnya di dalam database dan tombol add | | |
| 4 | Sistem harus bisa menampilkan persentase ujaran negatif | DMCF_0400 | Melihat persentase |
| 5 | Sistem harus bisa menampilkan data detail ujaran negatif yang tercatat secara keseluruhan oleh sistem. | DMCF_0500 | Melihat detail data |
| 5.1 | Sistem harus menyediakan sarana visual berupa diagram garis untuk menampilkan informasi data detail ujaran negatif yang berupa jumlah ujaran negatif serta waktu data tersebut diambil | DMCF_0501 | |
| 6 | Sistem harus bisa menampilkan data detail ujaran negatif berdasarkan tanggal yang dipilih pengguna. | DMCF_0600 | Melihat detail data tanggal pilihan |
| 7 | Sistem harus bisa menampilkan data detail ujaran negatif pada tanggal saat sistem digunakan. | DMCF_0700 | Melihat detail data hari ini |
| 8 | Sistem harus bisa menampilkan data detail ujaran negatif pada satu hari sebelumnya saat sistem digunakan. | DMCF_0800 | Melihat detail data kemarin |
| 9 | Sistem harus bisa menampilkan seluruh data detail ujaran negatif yang tersimpan kedalam bentuk file workbook | DMCF_0900 | Export data ke excel |
| 9.1 | Tipe file workbook yang digunakan berupa file dengan ekstensi .xls | DMCF_0901 | |
| 10 | Sistem harus bisa menampilkan seluruh data detail ujaran negatif yang tersimpan berdasarkan kota yang dipilih pengguna kedalam bentuk file workbook | DMCF_1000 | Export data ke excel pada kota pilihan |
| 10.1 | Tipe file workbook yang digunakan berupa file dengan ekstensi .xls | DMCF_1001 | |
| 11 | Sistem harus bisa menampilkan informasi urutan waktu dengan jumlah ujaran negatif terbanyak | DMCF_1100 | Melihat waktu ujaran negatif terbanyak |

| No | Kebutuhan Sistem | Kode | Nama Use Case |
|----|-------------------------------------------------------------------------------------------------------|-----------|---------------------------------------|
| 12 | Sistem harus bisa menampilkan informasi urutan kota dengan jumlah persentase ujaran negatif tertinggi | DMCF_1200 | Melihat kota ujaran negatif terbanyak |
| 13 | Sistem harus bisa menghentikan pengambilan data | DMCF_1300 | Menghentikan pengambilan data |

1.4 Daftar Kebutuhan Non Fungsional

Kebutuhan non fungsional adalah kebutuhan yang berhubungan dengan batasan serta kualitas dari sistem yang dibangun. Kebutuhan non fungsional juga menjadi syarat dalam pembangunan sebuah sistem. Kebutuhan non fungsional dari sistem *dashboard* monitoring *cyberbullying* ini berdasarkan kebutuhan fungsional yang sebelumnya telah didefinisikan serta berasal dari gambaran umum sistem yang nantinya akan dibangun. Setiap kebutuhan non fungsional diberikan kode DMCNF_XX, dengan xx sebagai penomoran dari kebutuhan tersebut. Daftar kebutuhan non fungsional dapat dilihat pada Tabel 4.3.

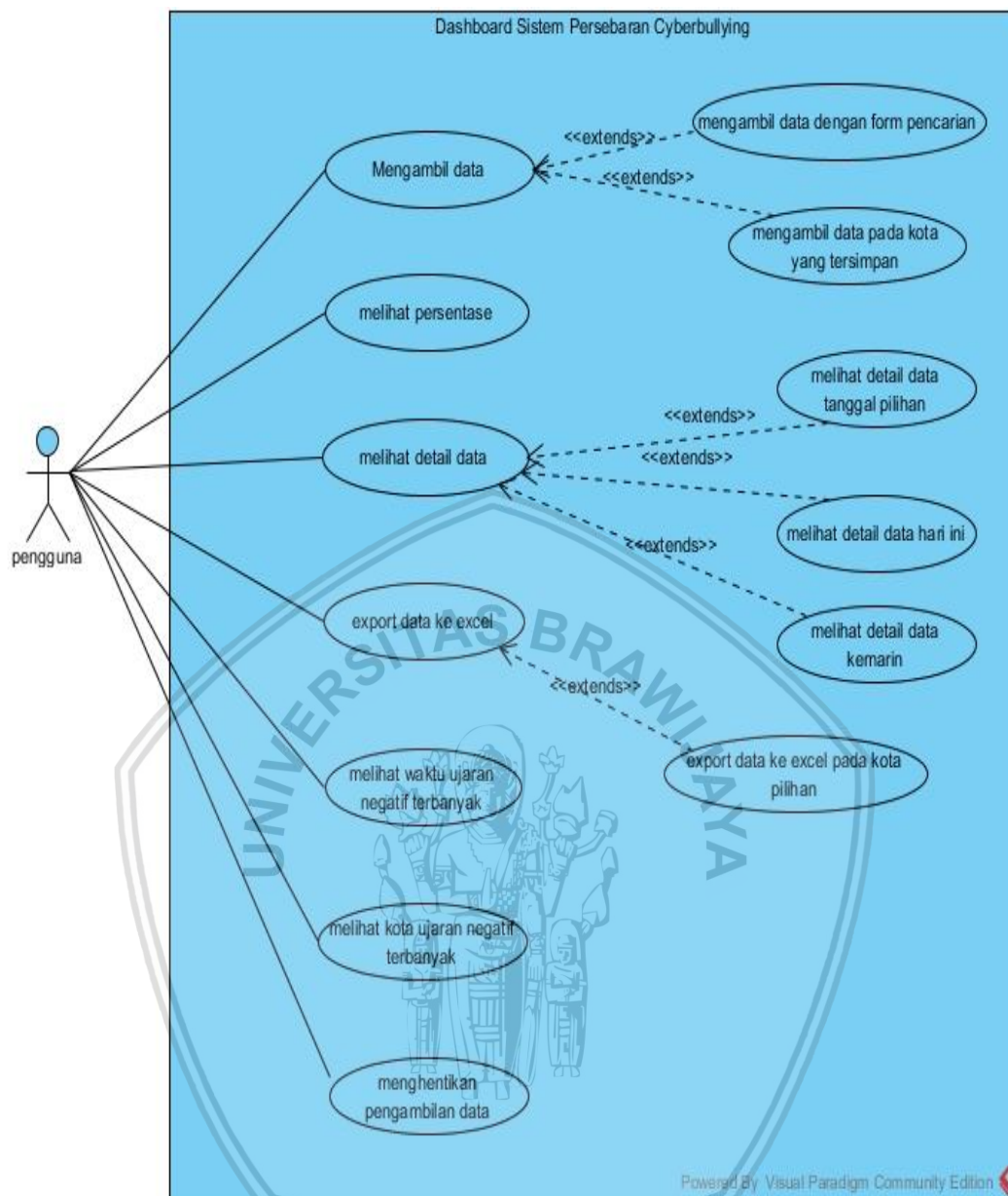
Tabel 0.2 Tabel Kebutuhan Non Fungsional Sistem

| Kode | Parameter | Spesifikasi |
|----------|----------------------|--------------------------------------------------------------------------------------------|
| DMCNF_01 | <i>Compatibility</i> | Sistem harus dapat dijalankan pada <i>browser Google Chrome</i> dan <i>Mozilla Firefox</i> |

1.5 Pemodelan Kebutuhan

1.5.1 Use Case Diagram

Use case diagram adalah diagram yang digunakan untuk menggambarkan perilaku sistem dari sudut pandang aktor. Pada *use case* diagram berisi semua kebutuhan sistem yang telah didefinisikan sebelumnya pada proses pengumpulan kebutuhan serta *use case* diagram menunjukkan interaksi yang dapat dilakukan stakeholder berdasarkan tujuan dari sistem yang dibangun. *Use case* diagram pada sistem ini dibuat berdasarkan kebutuhan fungsional yang telah ditetapkan dan dapat dilihat pada Gambar 4.1.



Gambar 0.1 Use Case Diagram

1.5.2 Use Case Scenario

Use case scenario merupakan tahapan kerja dari use case yang telah didefinisikan pada *use case diagram*. Pada *use case scenario* merupakan bentuk penjelasan detail terhadap setiap fungsi yang telah didefinisikan pada *use case diagram*. *Use case scenario* dibentuk didalam sebuah tabel yang berisi penjelasan dari use case name untuk mengetahui nama *use case*, actor untuk mengetahui siapa yang menggunakan, objective untuk mengetahui tujuan dari fungsi yang ingin dicapai, *pre condition* untuk mengetahui keadaan awal sebelum sebuah fungsi dijalankan, *basic flow* untuk mengetahui langkah atau tahapan, *alternative flow* untuk mengetahui keadaan ketika tidak sesuai tujuan, serta *post condition* untuk mengetahui keadaan akhir setelah fungsi dijalankan.

1.5.2.1 Mengambil Data

Tabel 0.3 *Use Case Scenario* Mengambil Data

| | |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name | Mengambil data |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat mengambil data secara berkala pada interval waktu tertentu serta pada lokasi yang dipilih dengan melakukan klik pada map dan menunjukkan tanda nya dengan <i>marker</i> |
| Precondition | Pengguna telah membuka halaman utama dan <i>browser</i> telah meload peta |
| Basic flow | 1. Pengguna menekan klik kiri pada lokasi yang terdapat di dalam maps |
| Alternative flow | 1. Mengambil data dengan form pencarian 2. Mengambil data pada kota yang tersimpan |
| Post Condition | Sistem akan memulai proses pengambilan data untuk setiap interval waktu tertentu pada lokasi yang telah ditentukan pengguna yang ditunjukkan dengan <i>marker</i> |

1.5.2.2 Mengambil Data Dengan Form Pencarian

Tabel 0.4 *Use Case Scenario* Mengambil Data Dengan Form Pencarian

| | |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name | Mengambil data dengan form pencarian |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat mengambil data secara berkala pada interval waktu tertentu serta pada lokasi tertentu dengan menginputkan nama lokasi yang ditentukan pengguna |
| Precondition | Pengguna telah membuka halaman <i>home</i> dan browser telah meload peta. |
| Basic flow | 1. Mengklik menu <i>search</i> 2. Pengguna memasukkan nama kota pada textfield 3. Pengguna menekan tombol yang memiliki <i>icon search</i> |
| Alternative flow | 1. Sistem akan menampilkan alert ketika lokasi yang dimasukkan tidak berhasil ditemukan |
| Post Condition | Sistem akan memulai proses pengambilan data untuk setiap interval waktu tertentu pada lokasi yang diinputkan pengguna yang ditunjukkan dengan <i>marker</i> |

1.5.2.3 Mengambil Data Pada Kota Yang Tersimpan

Tabel 0.5 Use Case Scenario Mengambil Data Pada Kota Yang Tersimpan

| | |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name | Mengambil data pada kota yang tersimpan |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat mengambil data secara berkala pada interval waktu tertentu serta pada kota tertentu dengan memilih daftar kota yang tersedia, yang sebelumnya telah tersimpan di dalam sistem |
| Precondition | Pengguna telah membuka halaman <i>home</i> dan browser telah meload peta. |
| Basic flow | <ol style="list-style-type: none"> 1. Pengguna mengklik menu <i>recently added</i> 2. Pengguna memilih list kota yang tersedia 3. Pengguna menekan tombol yang memiliki <i>icon</i> tambah |
| Alternative flow | - |
| Post Condition | Sistem akan memulai proses pengambilan data untuk setiap interval waktu tertentu pada lokasi yang telah dipilih pengguna yang ditunjukkan dengan <i>marker</i> |

1.5.2.4 Melihat Persentase

Tabel 0.6 Use Case Scenario Melihat Persentase

| | |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name | Melihat persentase |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat melihat persentase ujaran negatif dari lokasi yang dipilih |
| Precondition | Pengguna telah memilih lokasi yang sebelumnya terdapat proses pengambilan data yang ditunjukkan oleh <i>marker</i> |
| Basic flow | <ol style="list-style-type: none"> 1. Pengguna melakukan klik kiri pada <i>marker</i> |
| Alternative flow | - |
| Post Condition | Pada bagian atas <i>marker</i> akan menampilkan info konten yang berisi persentase ujaran negatif, serta persentase ujaran negatif akan ditampilkan dengan diagram <i>pie</i> pada bagian persentase <i>cyberbullying</i> |

1.5.2.5 Melihat Detail Data

Tabel 0.7 Use Case Scenario Melihat Detail Data

| | |
|----------------------|-----------------------------------------------------------------------------|
| Use Case Name | Melihat detail data |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat melihat seluruh detail data ujaran negatif |

| | |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Precondition | Pengguna telah memilih lokasi yang sebelumnya terdapat proses pengambilan data yang ditunjukkan oleh <i>marker</i> |
| Basic flow | 1. Pengguna melakukan klik kiri pada <i>marker</i> atau menekan button lihat seluruh data |
| Alternative flow | <ol style="list-style-type: none"> 1. Melihat detail data dengan tanggal yang tersedia pada kota pilihan 2. Melihat detail data hari ini atau pada hari saat sistem digunakan 3. Melihat detail data pada satu hari sebelumnya saat sistem digunakan |
| Post Condition | Pada bagian diagram garis jumlah ujaran negatif, akan menampilkan informasi mengenai seluruh data ujaran negatif beserta waktunya, yang telah tersimpan di dalam database serta ditunjukkan dengan menggunakan diagram garis |

1.5.2.6 Melihat Detail Data Tanggal Pilihan

Tabel 0.8 *Use Case Scenario* Melihat detail data tanggal pilihan

| | |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name | Melihat detail data tanggal pilihan |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat melihat detail data berdasarkan tanggal yang dipilih pengguna |
| Precondition | Pengguna telah melakukan klik kiri pada <i>marker</i> pada lokasi yang dituju |
| Basic flow | 1. Pengguna memilih menu <i>dropdown</i> tanggal yang tersedia pada lokasi yang dipilih sebelumnya |
| Alternative flow | - |
| Post Condition | Pada bagian diagram garis jumlah ujaran negatif, akan menampilkan informasi mengenai seluruh data ujaran negatif yang tersimpan di dalam database berdasarkan tanggal yang dipilih pengguna yang divisualisasikan dengan menggunakan diagram garis |

1.5.2.7 Melihat Detail Data Hari Ini

Tabel 0.9 *Use Case Scenario* Melihat Detail Data Hari ini

| | |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name | Melihat detail data hari ini |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat melihat detail dari keseluruhan data ujaran negatif yang tersimpan didalam database pada hari saat pengguna mengakses sistem |

| | |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Precondition | Pengguna telah melakukan klik kiri pada <i>marker</i> pada lokasi yang dituju |
| Basic flow | 1. Pengguna mengklik tombol data hari ini |
| Alternative flow | - |
| Post Condition | Pada bagian diagram garis jumlah ujaran negatif, akan menampilkan informasi mengenai seluruh data ujaran negatif yang tersimpan di dalam database berdasarkan tanggal saat pengguna mengakses sistem yang divisualisasikan dengan menggunakan diagram garis |

1.5.2.8 Melihat Detail Data Kemarin

Tabel 0.10 Use Case Scenario Melihat Detail Data Kemarin

| | |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name | Melihat detail data kemarin |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat melihat detail dari keseluruhan data ujaran negatif yang tersimpan di dalam database pada satu hari sebelumnya saat pengguna mengakses sistem |
| Precondition | Pengguna telah melakukan klik kiri pada <i>marker</i> pada lokasi yang dituju |
| Basic flow | 1. Pengguna mengklik tombol data 1 hari lalu |
| Alternative flow | - |
| Post Condition | Pada bagian diagram garis jumlah ujaran negatif, akan menampilkan informasi mengenai seluruh data ujaran negatif yang tersimpan di dalam database pada satu hari sebelumnya saat pengguna mengakses sistem yang divisualisasikan dengan menggunakan diagram garis |

1.5.2.9 Export Data ke Excel

Tabel 0.11 Use Case Scenario Export Data ke Excel

| | |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name | Export data ke excel |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat menyimpan keseluruhan detail data yang tersimpan di database kedalam file workbook yang berekstensi .xls |
| Precondition | Pengguna telah membuka halaman utama dan browser telah meload peta |
| Basic flow | <ol style="list-style-type: none"> 1. Pengguna memilih menu dengan <i>icon save</i> pada navbar 2. Pengguna mengklik <i>export data ke xls</i> |

| | |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Alternative flow | 1. <i>Export</i> data ke excel pada kota pilihan |
| Post Condition | Browser akan mengunduh dan menyimpan file <i>workbook</i> dengan ekstensi .xls yang berisi seluruh data ujaran negatif, positif, serta netral yang tersimpan di dalam database |

1.5.2.10 *Export Data Ke Excel Pada Kota Pilihan*

Tabel 0.12 Use Case Scenario Export Data Ke Excel Pada Kota Pilihan

| | |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name | <i>Export data kota pilihan</i> |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat menyimpan keseluruhan detail data berdasarkan kota yang dipilih pengguna kedalam bentuk excel |
| Precondition | Pengguna telah membuka halaman utama dan browser telah meload peta |
| Basic flow | 1. Pengguna memilih icon save pada kota yang tersedia di dalam bagian data tabel save kota pilihan ke excel |
| Alternative flow | - |
| Post Condition | Browser akan mengunduh dan menyimpan file <i>workbook</i> dengan ekstensi .xls yang berisi data ujaran negatif, positif, serta netral pada kota yang dipilih pengguna |

1.5.2.11 *Melihat Waktu Ujaran Negatif Terbanyak*

Tabel 0.13 Use Case Scenario Menampilkan Waktu Ujaran Negatif Terbanyak

| | |
|-------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| Use Case Name | Melihat waktu ujaran negatif terbanyak |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat melihat informasi urutan waktu dengan jumlah ujaran negatif terbanyak pada suatu kota |
| Precondition | Pengguna telah memilih kota yang ingin dituju dengan membuat <i>marker</i> pada area yang dituju |
| Basic flow | 1. Pengguna melakukan klik kiri pada <i>marker</i> |
| Alternative flow | - |
| Post Condition | Sistem akan menampilkan tiga waktu data dengan ujaran negatif terbanyak serta divisualisasikan dengan menggunakan diagram pie |

1.5.2.12 *Melihat Kota Ujaran Negatif Terbanyak*

Tabel 0.14 Use Case Scenario Melihat Kota Ujaran Negatif Terbanyak

| | |
|----------------------|---------------------------------------|
| Use Case Name | Melihat kota ujaran negatif terbanyak |
| Actors | Pengguna |

| | |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Objective | Mengizinkan pengguna untuk dapat melihat informasi urutan kota dengan jumlah persentase ujaran negatif tertinggi |
| Precondition | Pengguna telah membuka halaman home |
| Basic flow | 1. Pengguna memilih tombol update pada bagian rangking kota ujaran negatif terbanyak |
| Alternative flow | - |
| Post Condition | Pada halaman home di bagian rangking kota akan menampilkan urutan kota dengan ujaran negatif tertinggi yang divisualisasikan dengan menggunakan diagram garis |

1.5.2.13 Menghentikan Pengambilan Data

Tabel 0.15 Use Case Scenario Menghentikan Pengambilan Data

| | |
|-------------------------|------------------------------------------------------------------------------------------------------------------|
| Use Case Name | Menghentikan pengambilan data |
| Actors | Pengguna |
| Objective | Mengizinkan pengguna untuk dapat menghentikan proses pengambilan data pada lokasi yang ditunjuk <i>marker</i> |
| Precondition | Pengguna telah membuka halaman home dan sebelumnya telah terdapat proses pengambilan data pada lokasi tertentu |
| Basic flow | 1. Pengguna menekan klik kanan pada <i>marker</i> |
| Alternative flow | - |
| Post Condition | Proses pengambilan data akan berhenti yang ditunjukkan dengan hilangnya marker pada lokasi yang dipilih tersebut |

BAB 5 PERANCANGAN

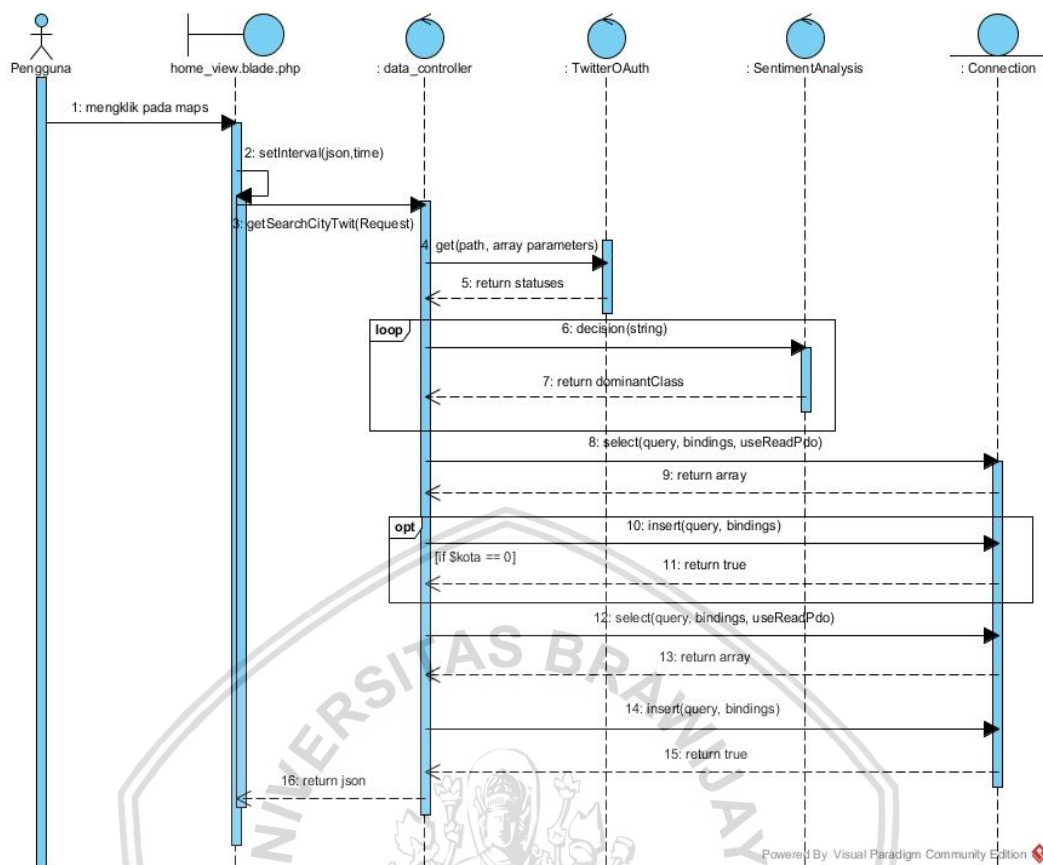
Perancangan merupakan sebuah proses untuk membangun model abstrak dari sistem, dengan setiap model mempresentasikan perspektif yang berbeda dari sistem. Tahapan perancangan pada umumnya juga sering disebut pemodelan. Perancangan ini digunakan juga untuk merepresentasikan sebuah sistem yang dibangun dengan notasi *Unified Modelling Language* (UML). Proses perancangan dalam sistem *dashboard* monitoring persebaran *cyberbullying* ini dibagi menjadi beberapa tahap yaitu perancangan *sequence* diagram, perancangan *class* diagram, perancangan komponen, perancangan basis data dan perancangan antar muka.

1.1 Perancangan *Sequence* diagram

Sequence diagram merupakan diagram yang menunjukkan interaksi antara aktor dengan sistem serta antara komponen-komponen didalam sistem. Didalam *sequence* diagram juga menjelaskan urutan-urutan proses interaksi dalam mencapai sebuah fungsionalitas. Komponen-komponen didalam sistem yang digambarkan didalam *sequence* diagram umumnya disebut objek, dimana objek-objek ini didapatkan dari hasil identifikasi tahap analisis kebutuhan, sehingga terdapat tiga *sequence* diagram yang menyusun sistem ini.

1.1.1 Mengambil Data

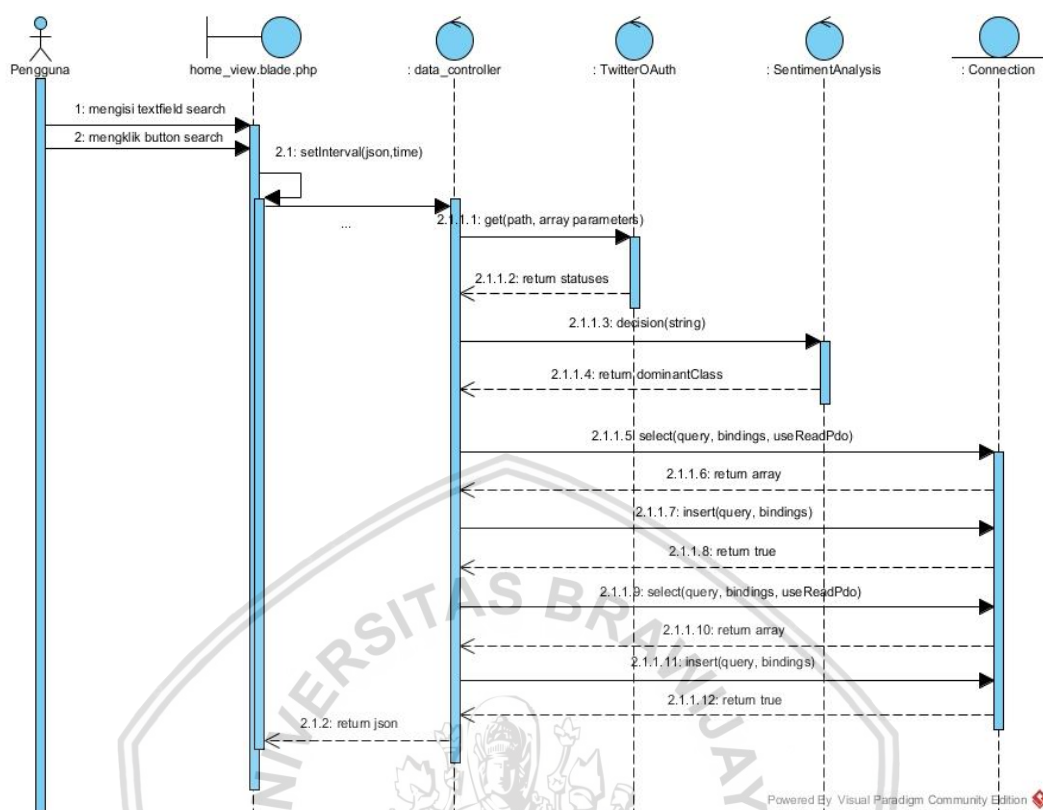
Sequence diagram mengambil data yang ditunjukkan pada Gambar 5.1 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Tahapan ini dimulai setelah sistem berhasil menampilkan halaman home dan menampilkan maps, pengguna melakukan klik kiri pada kota yang akan diambil datanya yang tersedia pada maps, lalu `home_view` memanggil fungsi `setInterval` javascript yang memanggil `data_controller` dengan method `getSearchCityTwit` dengan parameter `Request`. Setelah itu controller akan memanggil `TwitterOAuth` sebagai perantara dengan `Twitter API` dalam method `get` untuk mengambil data. Setelah data didapatkan data diklasifikasikan dengan menggunakan `SentimentAnalysis` dengan memanggil method `decision` dan data di looping sebanyak array data yang tersedia, setelah itu memanggil fungsi `select` untuk mengambil nama kota dari db, jika kota belum tersimpan maka memanggil fungsi `insert` dari class `Connection` untuk menginsertkan nama kota tersebut. Setelah itu `data_controller` memanggil fungsi `select` dari kelas `Connection` untuk memanggil id dari kota yang diinsertkan sebelumnya dan setelah mengembalikan hasilnya, `data_controller` memanggil fungsi `insert` untuk menginsertkan data hasil `decision` pada suatu kota.



Gambar 0.1 Sequence Diagram Mengambil Data

1.1.2 Mengambil Data Dengan Form Pencarian

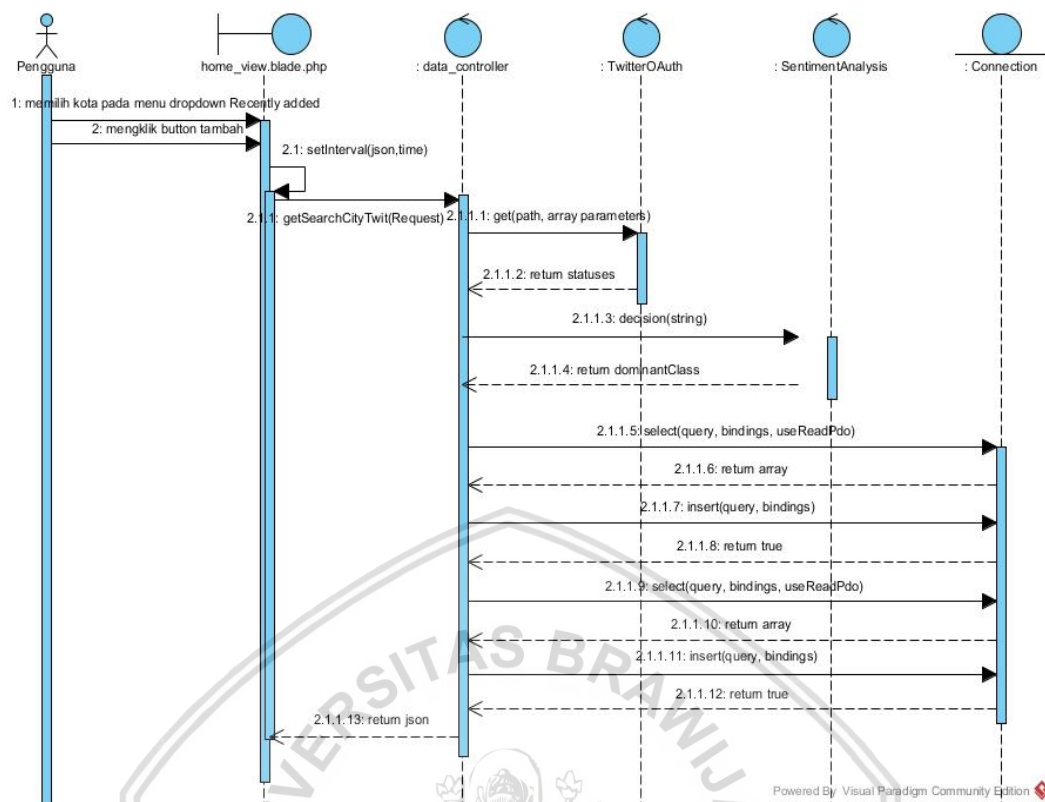
Sequence diagram mengambil data dengan form pencarian yang ditunjukkan pada Gambar 5.2 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Pertama setelah sistem berhasil menampilkan halaman home dan menampilkan maps, pengguna mengklik menu *search* kemudian akan tampil sebuah textfield. Setelah itu pengguna dapat mengisi lokasi yang ingin diambil datanya kemudian menekan tombol yang memiliki *icon search*, lalu *home_view* memanggil fungsi *set interval javascript* yang memanggil *data_controller* dengan method *getSearchCityTwit* dengan parameter *request* setelah itu *controller* akan memanggil *TwitterOAuth* sebagai perantara dengan *Twitter API* dalam method *get* untuk mengambil data. Setelah data didapatkan data diklasifikasikan dengan menggunakan *SentimentAnalysis* dengan memanggil method *decision* dan data di looping sebanyak array data yang tersedia, setelah itu memanggil fungsi *select* untuk mengambil nama kota dari db, jika kota belum tersimpan maka memanggil fungsi *insert* dari class *Connection* untuk menginsertkan nama kota tersebut. Setelah mengembalikan nilai *true*, *data_controller* memanggil fungsi *select* dari kelas *Connection* untuk memanggil *id* dari kota yang diinsertkan sebelumnya dan setelah mengembalikan hasilnya, *data_controller* memanggil fungsi *insert* untuk menginsertkan data hasil *decision* pada suatu kota.



Gambar 0.2 Sequence Diagram Mengambil Data Dengan Form Pencarian

1.1.3 Mengambil Data Pada Kota Yang Tersimpan

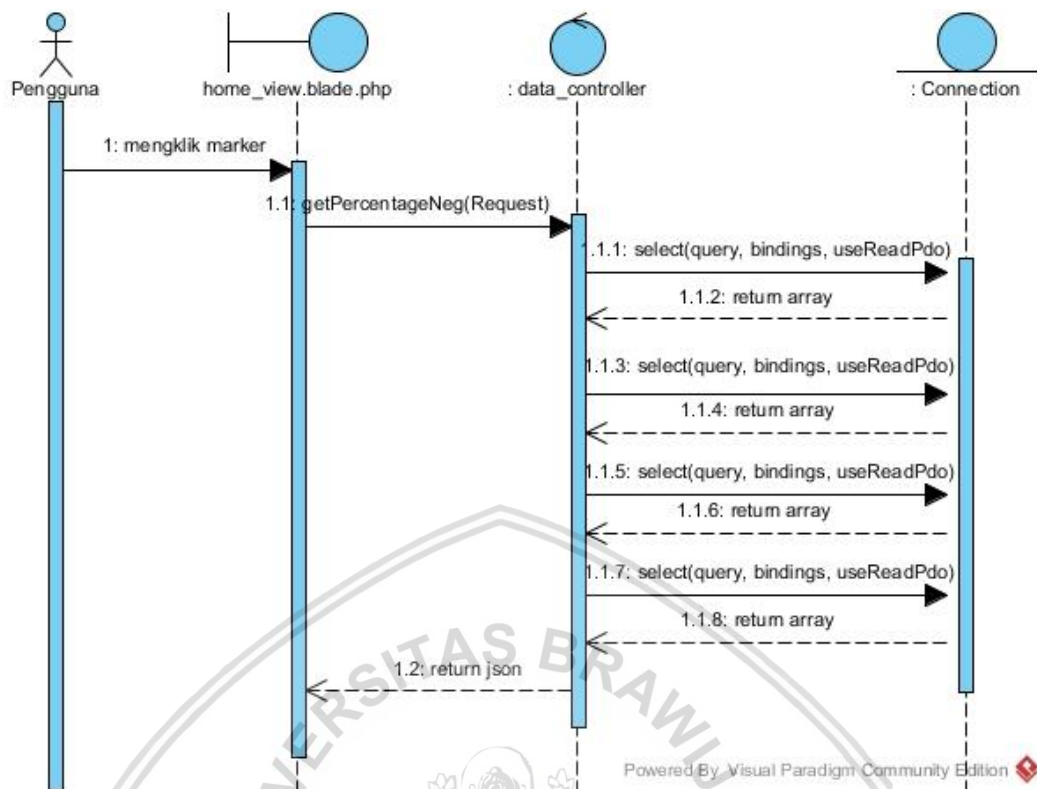
Sequence diagram mengambil data pada kota yang tersimpan ditunjukkan pada Gambar 5.3 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Pertama setelah sistem berhasil menampilkan halaman home dan menampilkan maps, pengguna mengklik menu *recently added* kemudian pengguna memilih kota yang disediakan pada dropdown dan menekan tombol tambah, lalu *home_view* memanggil fungsi *set interval* javascript yang memanggil *data_controller* dengan method *getSearchKotaTwit* dengan parameter request setelah itu controller akan memanggil *TwitterOAuth* sebagai perantara dengan Twitter API dalam method *get* untuk mengambil data. Setelah data didapatkan, data diklasifikasikan dengan menggunakan *SentimentAnalysis* dengan memanggil method *decision* dan data di looping sebanyak array data yang tersedia, setelah itu memanggil fungsi *select* untuk mengambil nama kota dari db, jika kota belum tersimpan maka memanggil fungsi *insert* dari class *Connection* untuk menginsertkan nama kota tersebut. Setelah mengembalikan nilai *true*, *data_controller* memanggil fungsi *select* dari kelas *Connection* untuk memanggil id dari kota yang diinsertkan sebelumnya dan setelah mengembalikan hasilnya, *data_controller* memanggil fungsi *insert* untuk menginsertkan data hasil *decision* pada suatu kota.



Gambar 0.3 Sequence Diagram Mengambil Data Pada Kota Yang Tersimpan

1.1.4 Melihat Persentase

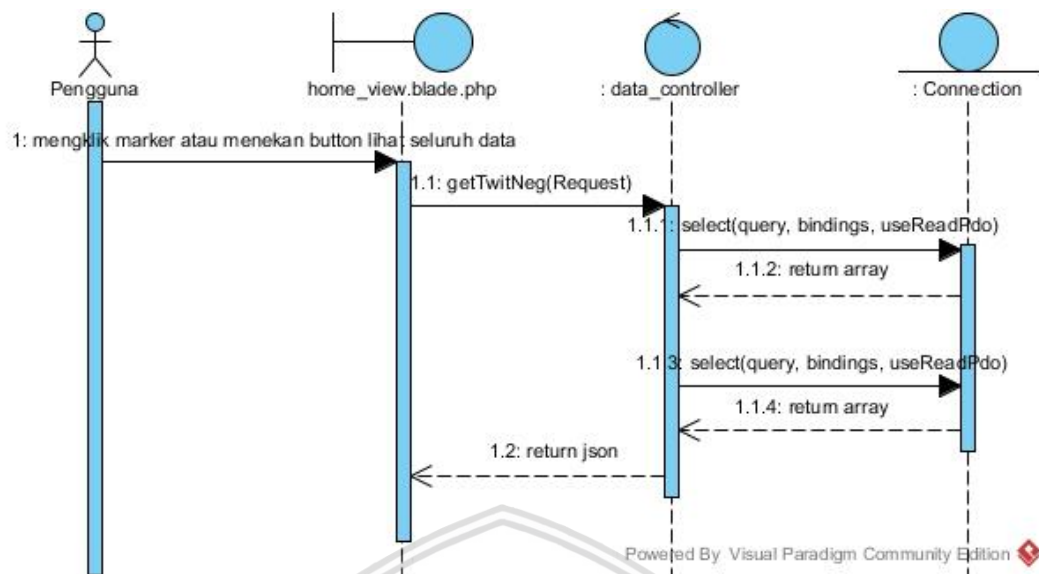
Sequence diagram melihat persentase pada kota pencarian yang ditunjukkan pada Gambar 5.4 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Tahapan ini dimulai Setelah *marker* muncul didalam peta, lalu pengguna mengklik *marker* untuk mendapatkan persentase, prosesnya yakni *home_view* akan memanggil method *getPercentageNeg* yang bertujuan untuk menghitung persentase negatif. Didalam method *getPercentageNeg* terdapat fungsi *select* yang pertama digunakan untuk mengambil id kota, setelah itu terdapat 3 fungsi *select* yang ketiganya digunakan untuk menjumlahkan ujaran negatif, positif, dan netral.



Gambar 0.4 Sequence Diagram Melihat Persentase

1.1.5 Melihat Detail Data

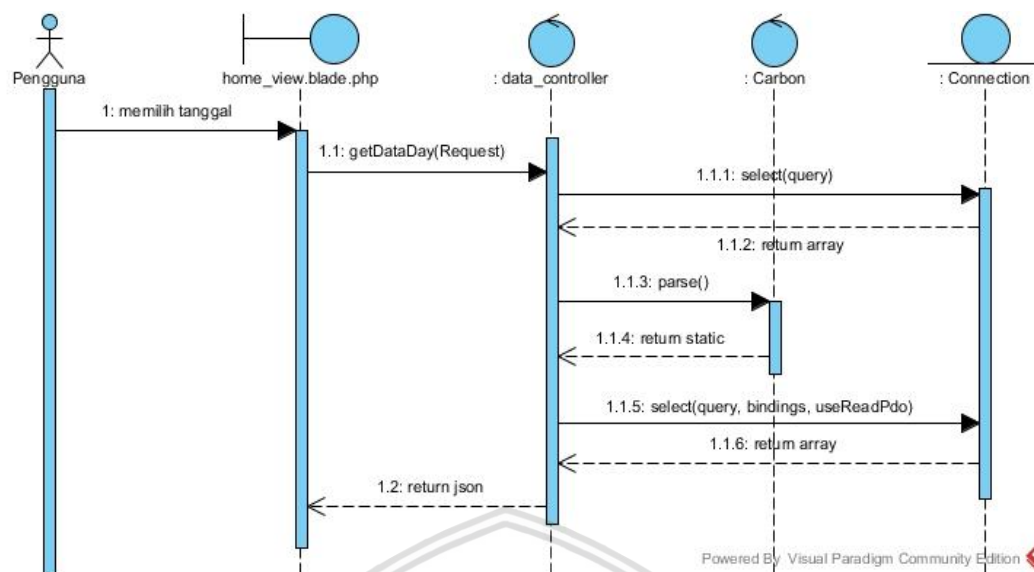
Sequence diagram melihat detail data pada kota pencarian yang ditunjukkan pada Gambar 5.5 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Tahapan ini dimulai dengan pengguna mengklik *marker* dari kota hasil pencarian yang ditemukan, lalu *home_view* akan memanggil method *getTwitNeg* pada *data_controller* yang berfungsi untuk mengambil seluruh ujaran negatif pada suatu kota dari database *dashboard*. Didalam method *getTwitNeg* terdapat fungsi *select* yang dipanggil dari kelas *Connection* dengan *select* pertama digunakan untuk mengambil id kota, dan *select* yang kedua digunakan untuk mengambil seluruh ujaran negatif dari kota yang diambil idnya tersebut.



Gambar 0.5 Sequence Diagram Melihat Detail Data

1.1.6 Melihat Detail Data Tanggal Pilihan

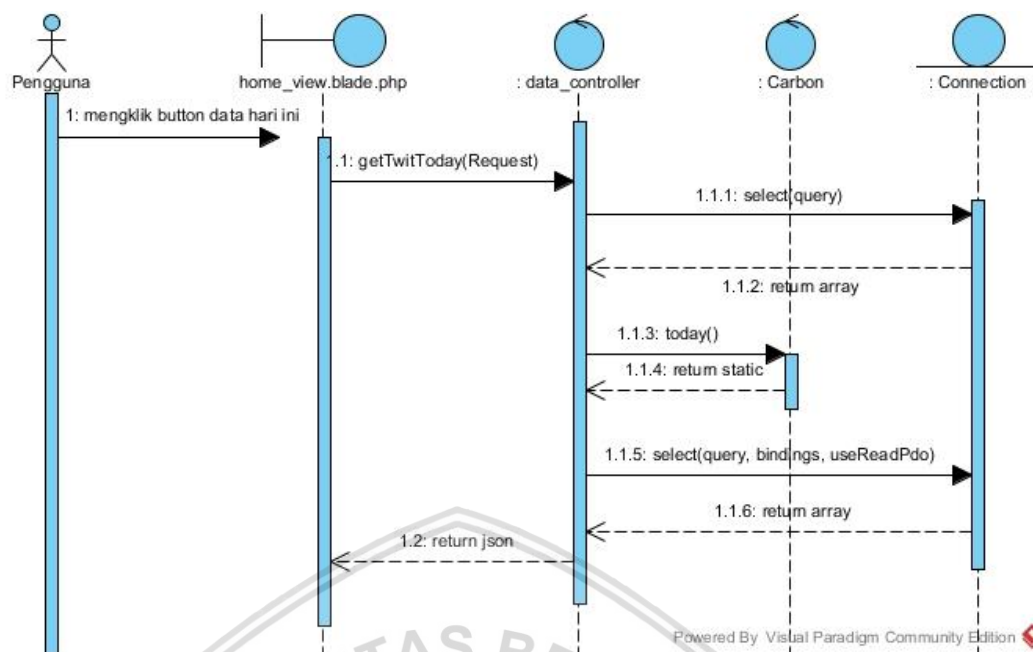
Sequence diagram melihat detail data tanggal pilihan yang ditunjukkan pada Gambar 5.6 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Fungsionalitas ini merupakan perluasan dari fungsionalitas melihat detail data. Pengguna dapat melihat tampilan *diagram* batang dengan seluruh data ujaran negatif berdasarkan tanggal yang tersimpan didalam database pada suatu kota tersebut. Pertama pengguna memilih tanggal yang tersedia pada suatu kota, setelah itu `home_view` akan memanggil `getDataDay` pada `data_controller`. Pada method `getDataDay` terdapat pemanggilan fungsi `select` dari kelas `Connection` yang digunakan untuk mengambil id dari kota yang di request pengguna, setelah mengembalikan hasil method `getDataDay` memanggil fungsi `parse` pada kelas `carbon` yang berfungsi untuk merubah format timestamp. Terakhir method `getDataDay` memanggil fungsi `select` yang digunakan untuk mengambil data ujaran negatif berdasarkan waktu dan id kota yang direquest.



Gambar 0.6 Sequence Diagram Melihat Detail Data Tanggal Pilihan

1.1.7 Melihat Detail Data Hari Ini

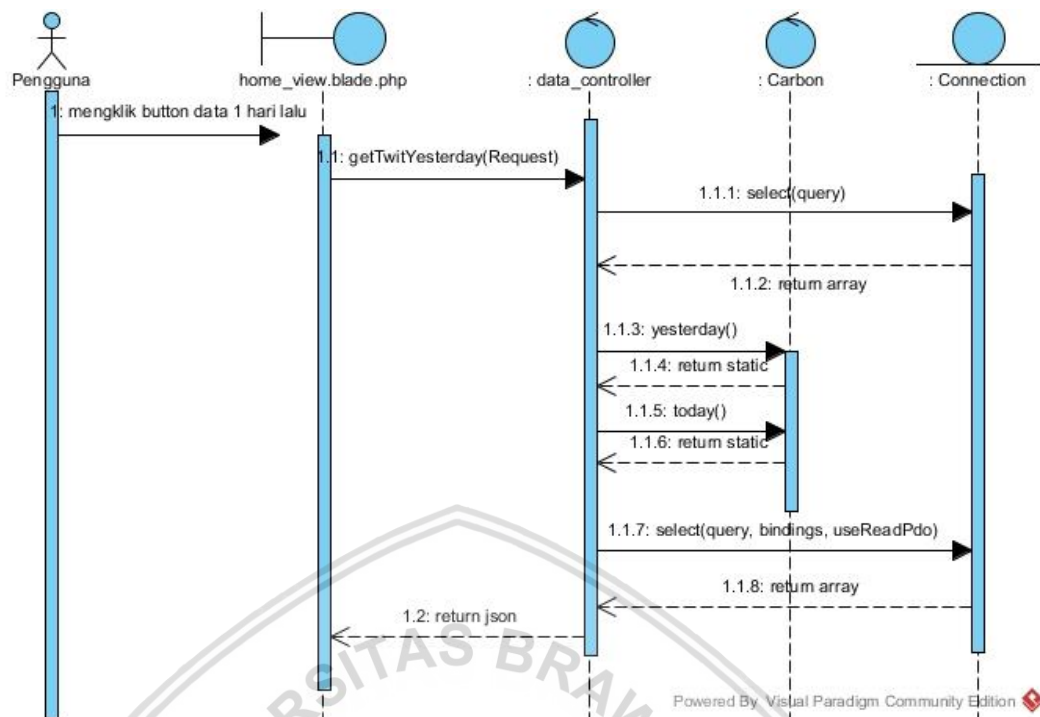
Sequence diagram melihat detail data hari ini yang ditunjukkan pada Gambar 5.7 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Fungsionalitas ini merupakan perluasan dari fungsionalitas melihat detail data. Pengguna dapat melihat tampilan *diagram* garis dengan seluruh data ujaran negatif berdasarkan hari saat pengguna mengakses sistem yang tersimpan di dalam database. Dalam melakukan fungsionalitas ini pengguna harus mengklik button data hari ini, setelah itu `home_view` akan memanggil `getTwitToday` pada `data_controller`. Pada method `getTwitToday` terdapat pemanggilan fungsi `select` dari kelas `Connection` yang digunakan untuk mengambil id dari kota yang di request pengguna. Setelah mengembalikan hasil, method `getDataDay` memanggil fungsi `today` pada kelas `Carbon` yang berfungsi untuk mengambil waktu hari ini yang menjadi syarat pada pemanggilan fungsi `select` pada kelas `Connection` yang dilakukan untuk mengambil data dengan id kota yang direquest pengguna dengan waktu ketika pengguna menggunakan sistem.



Gambar 0.7 Sequence Diagram Melihat Detail Data Hari Ini

1.1.8 Melihat Detail Data Kemarin

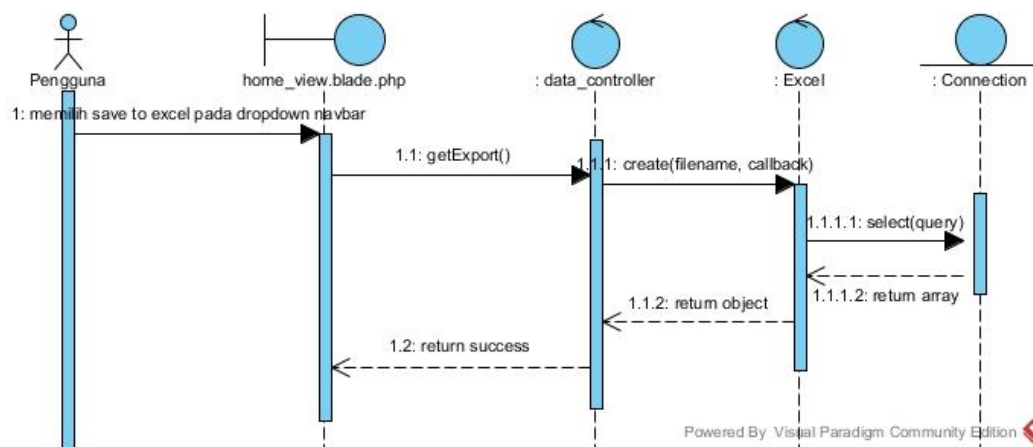
Sequence diagram melihat detail data kemarin yang ditunjukkan pada Gambar 5.8 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Fungsionalitas ini merupakan perluasan dari fungsionalitas melihat detail data. Pengguna dapat melihat tampilan *diagram* batang dengan seluruh data ujaran negatif berdasarkan satu hari sebelumnya saat pengguna mengakses sistem yang tersimpan di dalam database. Dalam melakukan fungsionalitas ini pengguna harus mengklik button data 1 hari lalu, setelah itu `home_view` akan memanggil `getTwitYesterday` pada `data_controller`. Pada method `getTwitYesterday` terdapat pemanggilan fungsi `select` dari kelas `Connection` yang digunakan untuk mengambil id dari kota yang di request pengguna. Setelah mengembalikan hasil, method `getDataDay` memanggil fungsi `today` dan `yesterday` pada kelas `Carbon` yang berfungsi untuk mengambil waktu hari ini yang menjadi syarat pada pemanggilan fungsi `select` pada kelas `Connection` yang dilakukan untuk mengambil data dengan id kota yang direquest pengguna dengan waktu satu hari sebelumnya ketika pengguna menggunakan sistem.



Gambar 0.8 Sequence Diagram Melihat Detail Data Kemarin

1.1.9 Export Data ke Excel

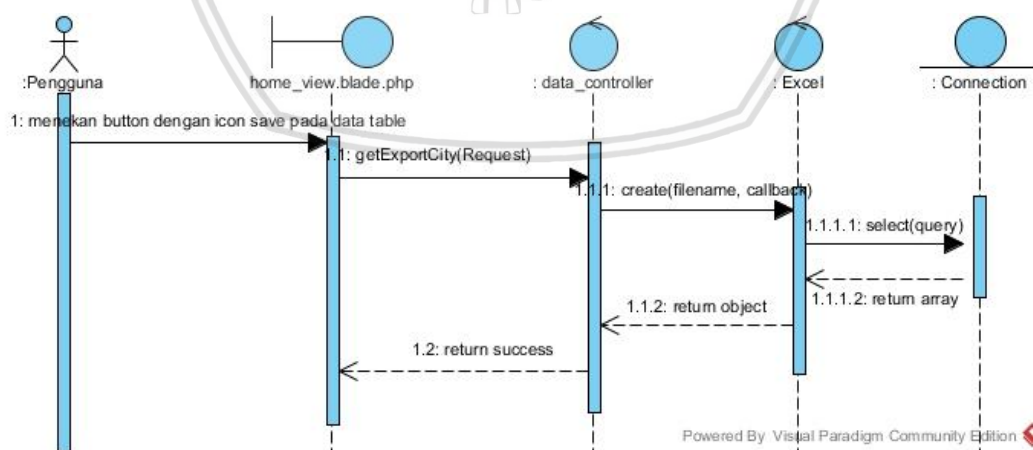
Sequence diagram export data ke excel yang ditunjukkan pada Gambar 5.9 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Tahapan ini dimulai dengan pengguna memilih menu option pada *dropdown* navbar dan memilih save to excel. Kemudian *home_view* akan ke *data_controller* dengan memanggil method *getExport* yang didalam method tersebut terdapat pemanggilan fungsi *create* pada kelas *Excel*. Didalam fungsi *create* terdapat pemanggilan fungsi *select* untuk memanggil seluruh data yang nantinya akan disimpan kedalam format excel.



Gambar 0.9 Sequence Diagram Export Data Ke Excel

1.1.10 Export Data Ke Excel Pada Kota Pilihan

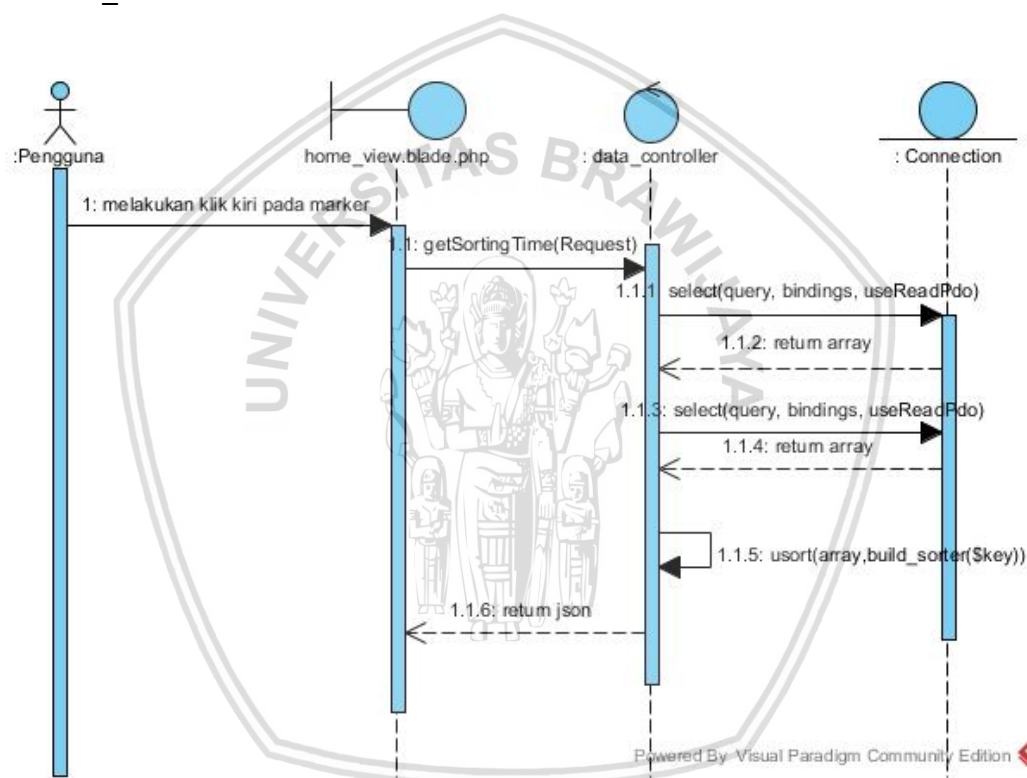
Sequence diagram export data ke excel pada kota pilihan yang ditunjukkan pada Gambar 5.10 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Fungsionalitas ini merupakan perluasan dari fungsionalitas *export data ke excel* hanya saja terdapat pada perbedaan query saat controller memanggil dengan menggunakan method *select*. Query yang digunakan spesifik ke kota tertentu berdasarkan pilihan pengguna. Tahapan ini dimulai dengan pengguna mengklik button pada kota yang tersedia pada data table. Kemudian *home_view* akan ke *data_controller* dengan memanggil method *getExportKota* yang didalam method tersebut terdapat pemanggilan fungsi *create* pada kelas *Excel*. Didalam fungsi *create* terdapat pemanggilan fungsi *select* untuk memanggil data berdasarkan kota yang direquest pengguna yang nantinya akan disimpan kedalam format excel.



Gambar 0.10 Sequence Diagram Export Data Ke Excel Pada Kota Pilihan

1.1.11 Melihat Waktu Ujaran Negatif Terbanyak

Sequence diagram melihat waktu ujaran negatif terbanyak yang ditunjukkan pada Gambar 5.11 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Tahapan ini dimulai dengan pengguna mengklik *marker* lalu pada *home_view* akan memanggil method *getSortingTime* yang terdapat pada *data_controller* dengan parameter kota yang dituju. Setelah itu controller akan memanggil fungsi *select* pada kelas *Connection* berdasarkan query yang tentukan setelah data didapatkan dari db lalu controller memanggil fungsi *usort* yang merupakan fungsi bawaan yang ditawarkan didalam php dengan parameter yaitu array yang telah didapatkan dari db dan fungsi *build sorter* berdasarkan key untuk melakukan sorting data dari terbesar hingga terkecil sebelum dikirimkan hasilnya ke *home_view*.

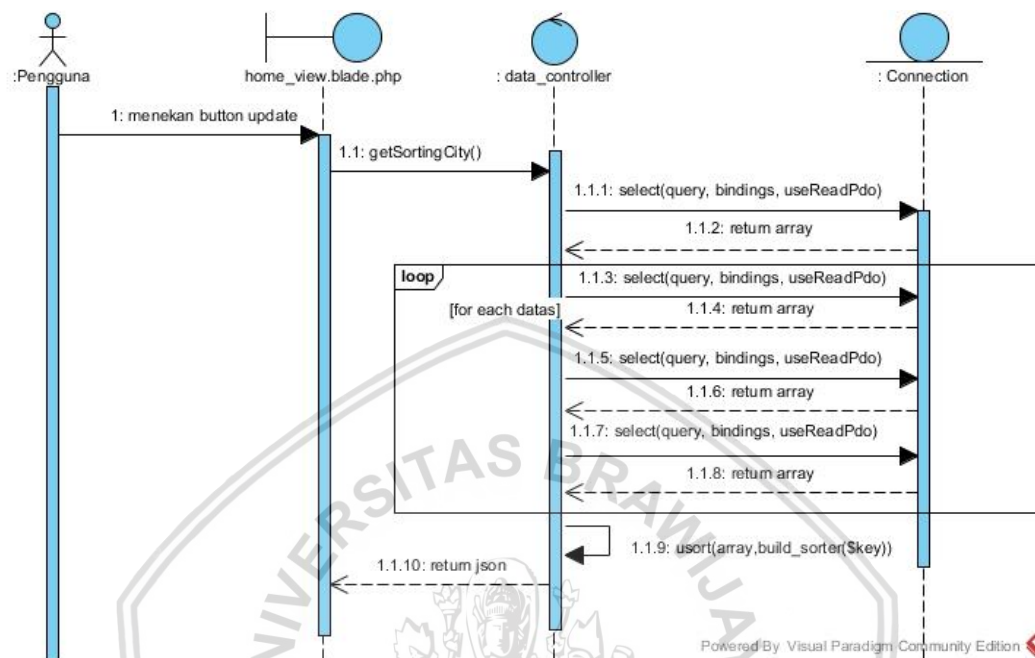


Gambar 0.11 *Sequence* Diagram Melihat Waktu Ujaran Negatif Terbanyak

1.1.12 Melihat Kota Ujaran Negatif Terbanyak

Sequence diagram melihat kota dengan ujaran negatif terbanyak ditunjukkan pada Gambar 5.12 menjelaskan tahapan dari sistem saat pengguna menjalankan fungsionalitas tersebut. Tahapannya dimulai saat pengguna memilih tombol update pada bagian urutan kota dengan ujaran negatif tertinggi. Lalu *home_view* akan memanggil *data_controller* dengan method *getSortingCity*. Setelah itu controller akan memanggil fungsi *select* untuk memanggil seluruh data berdasarkan kota yang tersimpan di db, dan tiga fungsi *select* yang dipanggil didalam perulangan *foreach* yang digunakan untuk mengambil serta menjumlahkan ujaran negatif, positif, maupun netral pada kota yang terdapat

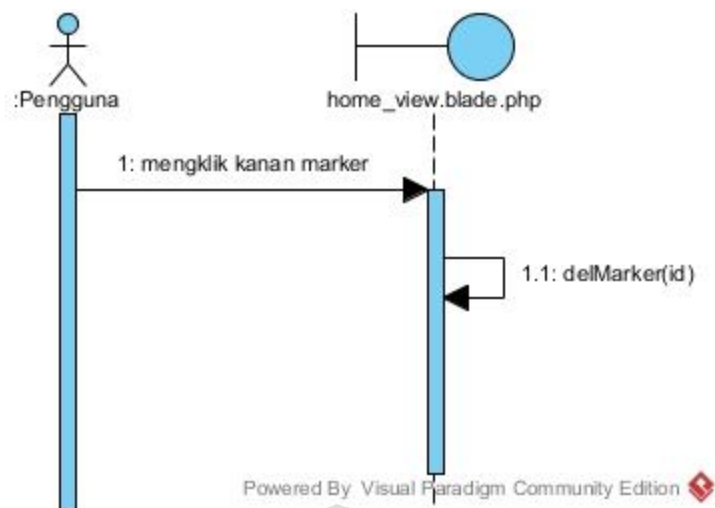
pada array datas hasil pemanggilan fungsi select yang pertama. Setelah itu controller memanggil fungsi bawaan php yaitu usort yang digunakan untuk melakukan pengurutan array.



Gambar 0.12 Sequence Diagram Melihat Kota Ujaran Negatif Terbanyak

1.1.13 Menghentikan Pengambilan Data

Sequence diagram menghentikan pengambilan data ditunjukkan pada Gambar 5.13. Tahapannya dimulai ketika sebelumnya telah terdapat proses pengambilan data pada suatu lokasi yang ditunjukkan oleh *marker*, *marker* tersebut sebelumnya telah diciptakan melalui klik pada maps, melakukan kota pencarian, dan memilih kota yang telah tersimpan. Tahapan ini diawali ketika pengguna mengklik kanan *marker*, lalu *home_view* akan memanggil fungsi javascript dengan nama `delMarker(id)` dan akan menghapus *marker* yang telah di create berdasarkan id dari *marker* yang di klik kanan.



Gambar 0.13 *Sequence Diagram* Menghentikan Pengambilan Data

1.2 Perancangan *Class Diagram*

Class diagram merupakan diagram yang menunjukkan objek dari kelas-kelas di dalam sebuah sistem dan hubungan antara kelas-kelas tersebut. Pada *class diagram* yang menyusun sistem *dashboard* monitoring persebaran *cyberbullying* menampilkan informasi mengenai atribut serta operasi secara detail yang menyusun tiap-tiap class. Pada perancangan *class diagram* ini juga berisi penggambaran notasi hubungan antar kelas. Perancangan *class diagram* akan dijelaskan pada Gambar 5.14 berikut.

Gambar 0.14 Class Diagram

Berdasarkan rancangan *class* diagram yang ditunjukkan pada Gambar 5.14 terdapat beberapa class yang menyusun fungsionalitas dari sistem. Kelas utama yang memproses seluruh fungsi dari *dashboard* terdapat pada kelas *data_controller*. kelas *TwitterOAuth* merupakan kelas yang berasal dari *library TwitterOAuth* yang digunakan untuk mengakses API Twitter. Kelas *Sentiment* dan *Sentiment Analysis* merupakan kelas yang berasal dari *library* analisis sentimen yang digunakan untuk melakukan klasifikasi. Kelas *carbon* merupakan kelas yang membantu dalam melakukan format timestamp didalam sistem. Sedangkan kelas *facade*, merupakan kelas yang terdapat pada *framework* yang digunakan yakni *framework* laravel yang berfungsi untuk mengatur dan menghubungkan antara *seviceprovider* kelas *db* dan *connection* yang digunakan untuk berhubungan dengan basis data yang digunakan pada sistem ini, serta kelas *excel* dengan *service provider* nya untuk melakukan *export data* ke dalam ekstensi *.xls*.

1.3 Perancangan Komponen

Perancangan komponen merupakan perancangan yang berisi algoritme dari tiap-tiap komponen yang menyusun sistem. Algoritme pada perancangan komponen bertujuan untuk memberikan gambaran serta memudahkan didalam melakukan pengkodean.

1.3.1 Algoritme Mengambil Data

Nama Kelas : DataController

Nama Operasi : *getSearchCityTwit(Request)*

Deskripsi : Algoritme ini digunakan ketika pengguna melakukan pencarian data ujaran negatif yang dapat dilakukan dengan cara mengklik lokasi kota pada maps, memasukkan lokasi pada form pencarian, ataupun berdasarkan kota yan telah tersimpan didalam database. Algoritme ini berisi pengambilan data dari Twitter API dan melakukan klasifikasi dengan analisis sentimen untuk nantinya disimpan ke dalam db dan digunakan ke dalam *dashboard*

Algoritme :

Tabel 0.1 Algoritme Mengambil Data

| Algoritme 1 : Fungsi mengambil data | |
|-------------------------------------|--------------------------------------------------------------------------------------|
| 1 | inisialisasi access token, access token secret |
| 2 | instansiasi class TwitterOAuth |
| 3 | inisialisasi content untuk memanggil method get untuk verify akun |
| 4 | dan mengambil data dari endpoint API Twitter |
| 5 | inisialisasi statuses untuk mengambil data dengan method get dari kelas TwitterOAuth |
| 6 | inisialisasi data = decode statuses |
| 8 | inisialisasi countpos, count neg, countneu = 0 |
| 9 | loop for dari i = 0 sampai jumlah data |
| 10 | instansiasi kelas SentimentAnalysis |

| Algoritme 1 : Fungsi mengambil data | |
|-------------------------------------|----------------------------------------------------------------------|
| 11 | inisialisasi array nilai = dilakukan penentuan positif, negatif atau |
| 12 | netral dengan memanggil method desicion pada kelas SentimentAnalysis |
| 13 | pada tweet ke-i |
| 14 | seleksi if jika nilai ke-i = negatif |
| 15 | countneg++ |
| 16 | else if nilai ke-i = positif |
| 17 | countpos++ |
| 18 | else |
| 19 | countneu++ |
| 20 | endif |
| 21 | inisialisasi nama kotas yaitu mengambil data dari tabel data_kota |
| 22 | dengan mengambil id, dan nama kota |
| 23 | inisialisasi kota = 0 |
| 24 | loop foreach nama kotas as nama kota |
| 25 | if nama kota pada request != nama kota |
| 26 | kota++ |
| 27 | endif |
| 28 | endforeach |
| 29 | if kota=0 |
| 30 | insert kota ke tabel data_kota |
| 31 | endif |
| 32 | inisialisasi id = mengambil id berdasarkan nama kota request dari |
| 33 | tabel data_kota |
| 34 | insert countpos,countneg,countneu kedalam data_jam bersama id |
| 35 | inisialisasi total |
| 36 | return total |

1.3.2 Algoritme Mencari Persentase Ujaran Negatif

Nama Kelas : DataController

Nama Operasi : getPercentageNeg(Request)

Deskripsi : Algoritme ini digunakan ketika pengguna mengklik *marker* untuk mendapatkan informasi mengenai persentase ujaran negatif didalam suatu kota. Didalamnya terdapat pengambilan data dari suatu kota dan perhitungan jumlah persentase untuk ujaran negatif dari kota yang dituju

Algoritme :

Tabel 0.2 Algoritme Mencari Persentase Ujaran Negatif

| Algoritme 2 : Fungsi mencari persentase ujaran negatif | |
|--------------------------------------------------------|-----------------------------------------------------------------------|
| 1 | public function getPercentageNeg(Request \$req){ |
| 2 | inisialisasi variable id = select id dari table data kota berdasarkan |
| 3 | nama kota yang diterima controller |
| 4 | percabangan if jika variabel id terdapat nilainya |
| 5 | var ids = id |
| 6 | inisialisasi variabel neg = select jumlah neg tabel data_jam dengan |
| 7 | id kota = ids |
| 8 | inisialisasi variabel pos = select jumlah pos tabel data_jam dengan |
| 9 | id kota = ids |
| 10 | inisialisasi variabel neu = select jumlah neu tabel data_jam dengan |
| 11 | id kota = ids |
| 12 | inisialisasi variabel Negatif = hasil query variable neg |
| 13 | inisialisasi variabel Positif = hasil query variable pos |
| 14 | inisialisasi variabel Neutral = hasil query variable neu |
| 15 | inisialisasi variabel total = Positif+Negatif+neutral |
| 16 | inisialisasi variable percent = (Negatif/total)*100; |
| 17 | |

| Algoritme 2 : Fungsi mencari persentase ujaran negatif | |
|--------------------------------------------------------|---------------------------------------------------------------------|
| 18 | inisialisasi hasil = format dua angka dibalekang koma dari variable |
| 19 | percent |
| 20 | return json_encode(hasil) |
| 21 | endif |
| 22 | else |
| 23 | return json_encode(null) |
| 24 | end ifelse |
| | end |

1.3.3 Algoritme Mengambil Data 1 Hari Lalu

Nama Kelas : DataController

Nama Operasi : getTwitYesterday (Request)

Deskripsi : Algoritme ini digunakan ketika pengguna mengklik lihat data 1 hari lalu pada *dashboard*. Didalam algoritme ini terdapat pengambilan data dengan query yang menyeleksi data berdasarkan waktu data diambil, yakni 1 hari sebelum pengguna menggunakan fungsi ini

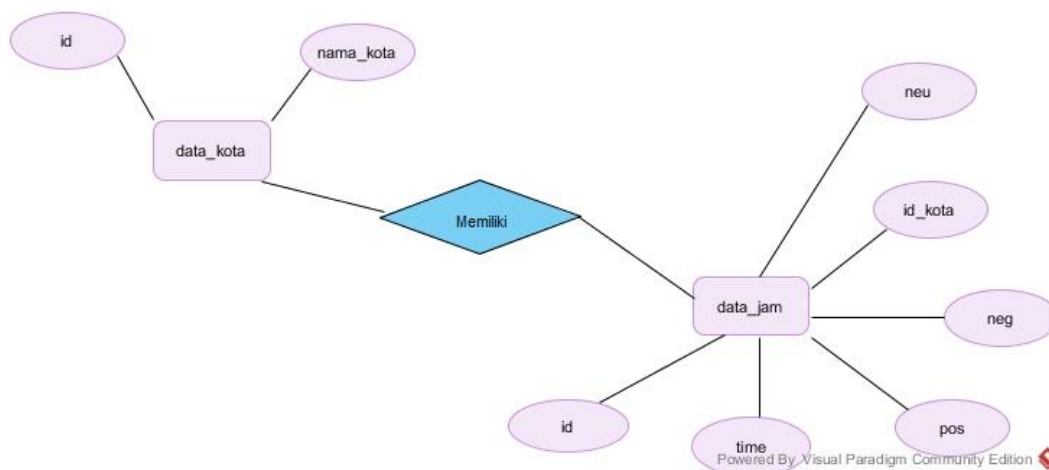
Algoritme :

Tabel 0.3 Algoritme Mengambil Data 1 Hari Lalu

| Algoritme 3 : Fungsi mengambil data 1 hari lalu | |
|-------------------------------------------------|-----------------------------------------------------------------------|
| 1 | public function getTwitYesterday(Request \$req){ |
| 2 | inisialisasi variable id = select id dari table data kota berdasarkan |
| 3 | nama kota yang diterima controller |
| 4 | inisialisasi variable ids = hasil query dr id |
| 5 | inisialisasi variable data = select neg, dan time berdasarkan ids |
| 6 | time > yesterday, time < today |
| 7 | percabangan if jika variable data diset |
| 8 | return json_encode(data); |
| 9 | else |
| 10 | return json_encode(null); |
| 11 | end if |
| 12 | end |

1.4 Perancangan Basis Data

Pada pembangunan sistem monitoring persebaran *cyberbullying* ini digunakan basis data untuk dapat menyimpan data yang diperlukan di dalam mencapai tujuan fungsionalitas sistem. Perancangan basis data ini bertujuan untuk memberikan gambaran dalam membangun basis data yang nantinya menyusun sistem, serta menggambarkan objek-objek beserta atribut apa saja yang diperlukan didalam membangun sistem ini. Dalam memodelkan data, tahapan yang dilakukan adalah mendefinisikan *Conceptual Data Model* serta *Physical Data Model*. *Conceptual Data Model* merupakan tahapan yang melibatkan identifikasi entitas serta hubungan yang mendefinisikan ruang lingkup masalah yang ditangani oleh sistem, sedangkan *Physical Data Model* merupakan tahapan untuk mentransformasikan hasil identifikasi entitas yang telah dilakukan pada *Conceptual Data Model* ke dalam tabel database fisik yang terperinci. Pada proses perancangan basis data, dilakukan tahapan *Conceptual Data Model* yang di notasikan dengan *Entity Relationship Diagram*.



Gambar 0.15 Conceptual Data Model Sistem Dashboard Monitoring Cyberbullying

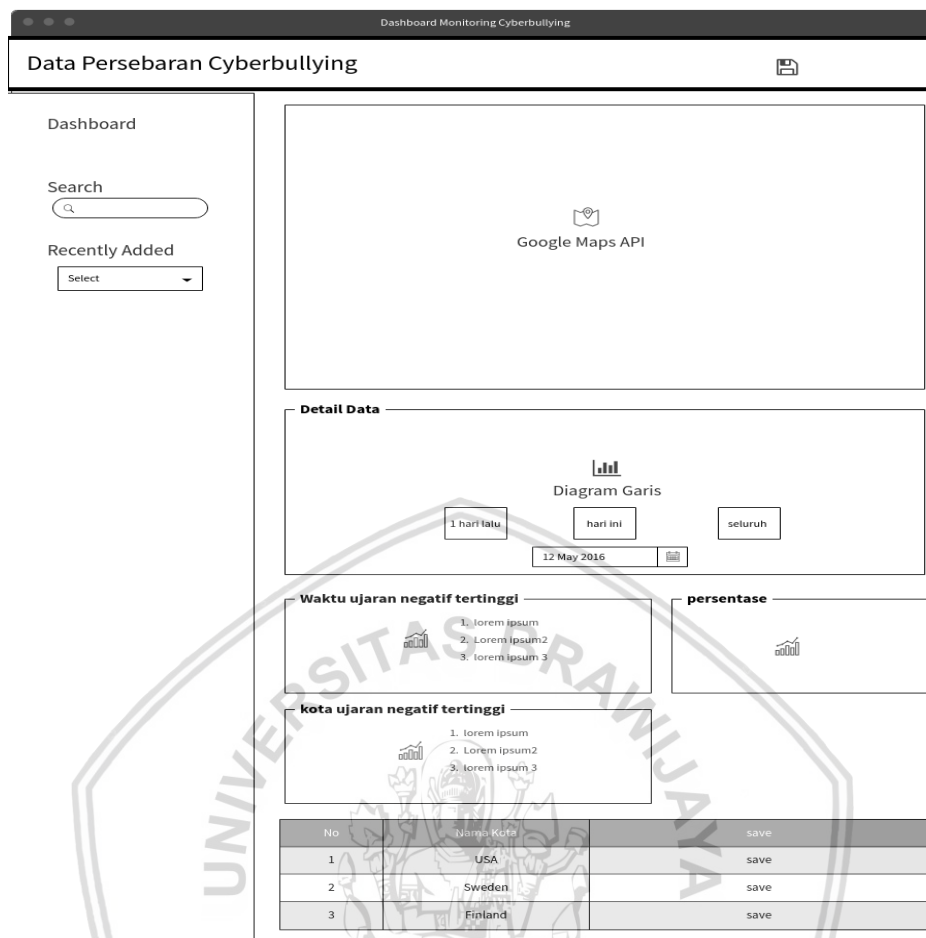
Dalam perancangan basis data yang digunakan didalam pembangunan sistem ini, terdapat dua entitas atau objek yang menyusun, yaitu *data_kota* dan *data_jam*. Data kota merepresentasikan objek kota yang nantinya diperlukan didalam implementasi. Data jam merepresentasikan objek data yang terdapat di dalam suatu kota. Data kota dan data jam memiliki hubungan one to many, yang artinya sebuah objek kota dapat memiliki banyak objek data.

1.5 Perancangan Antar Muka

Perancangan Antarmuka merupakan perancangan yang digunakan untuk menunjukkan halaman web yang menyusun sistem yang akan dibuat. Berikut adalah tampilan perancangan antarmuka sistem *dashboard* monitoring persebaran *cyberbullying*.

1.5.1 Halaman Home

Perancangan halaman home merupakan halaman utama dimana pengguna dapat memantau persebaran *cyberbullying* serta dapat memilih kota pemantauan.



Gambar 0.16 Halaman Home

Gambar 5.16 merupakan perancangan dari antarmuka halaman utama (home). Didalamnya terdapat beberapa bagian yang menginformasikan data secara visual. Pertama terdapat bagian maps dimana pengguna dapat memilih kota yang ingin dipantau dengan melakukan klik pada kota tersebut. Dibagian bawah maps terdapat bagian detail data, pada bagian ini terdapat diagram garis yang menunjukkan jumlah ujaran negatif pada suatu kota pada saat *marker* di klik. Kemudian di bagian bawah berturut-turut terdapat bagian waktu ujaran negatif tertinggi dan kota ujaran negatif tertinggi. Bagian ini menginformasikan kepada pengguna rentang waktu dengan ujaran negatif tertinggi suatu kota dan menampilkan urutan kota dengan ujaran negatif tertinggi. Pada bagian kanan waktu ujaran negatif tertinggi terdapat bagian persentase, bagian ini menampilkan persentase ujaran negatif suatu kota, selain dari info bar yang muncul diatas *marker*. Pada bagian paling bawah terdapat data tabel dimana pengguna dapat melakukan export data berdasarkan kota yang dipilih. Pada bagian kiri halaman home terdapat text box dan select box, text box digunakan sebagai masukkan pencarian kota, sedangkan select box sebagai masukan kota yang telah tersimpan sebelumnya.

BAB 6 IMPLEMENTASI

Pada bagian ini dilakukan tahapan implementasi dari sistem yang telah dirancang pada bab sebelumnya. Proses implementasi ini merupakan tahapan selanjutnya dari proses perancangan dimana didalamnya terdapat proses untuk membangun sistem berdasarkan perancangan yang telah ditetapkan. Tahapan didalam implementasi yakni mendefinisikan lingkungan perangkat keras dan lingkungan perangkat lunak yang digunakan, mendefinisikan implementasi kode program, implementasi basis data dan implementasi antarmuka sesuai dengan hasil perancangan.

1.1 Spesifikasi Perangkat Keras

Dalam pengembangan sistem *dashboard* monitoring *cyberbullying*, digunakan komputer dengan spesifikasi yang dijelaskan pada Tabel 6.1.

Tabel 0.1 Spesifikasi Perangkat Keras

| Nama Komponen | Spesifikasi |
|---------------------|------------------------------------------------------------|
| <i>System Model</i> | Lenovo |
| <i>Processor</i> | Intel(R) Core(TM) i3 CPU M 330 @ 2.13GHz (4 CPUs), ~2.1GHz |
| <i>Memory</i> | 6144 MB |
| <i>Display</i> | Intel(R) HD Graphics |
| <i>Hardisk</i> | 500 GB |

1.2 Spesifikasi Perangkat Lunak

Dalam melakukan pengembangan sistem *dashboard* monitoring *cyberbullying*, digunakan perangkat lunak dengan spesifikasi yang dijelaskan pada Tabel 6.2.

Tabel 0.2 Spesifikasi Perangkat Lunak

| Nama Komponen | Spesifikasi |
|-----------------------------------|---------------------------|
| <i>Operating System</i> | Windows 7 Ultimate 64-bit |
| <i>Programming language</i> | PHP, Javascript, HTML |
| <i>Programming Environment</i> | Sublime Text |
| <i>Database Management System</i> | - |

1.3 Batasan Implementasi

Terdapat batasan-batasan didalam proses implementasi sistem *dashboard* monitoring *cyberbullying*, yakni sebagai berikut:

1. Aplikasi dibangun berbasis web yang menggunakan *framework laravel* dengan menggunakan bahasa pemrograman PHP dan *Javascript*.
2. Pembuatan antarmuka aplikasi menggunakan HTML, CSS.
3. Implementasi basis data dibangun menggunakan DBMS mysql

1.4 Implementasi Kode Program

Pada implementasi kode program dilakukan berdasarkan perancangan komponen yang berisi algoritma-algoritma yang menyusun sistem yang dibangun. Tahapan ini menunjukkan perubahan algoritme pada perancangan komponen menjadi bahasa pemrograman yang dapat di *compile* oleh komputer. Implementasi kode program akan ditunjukkan dengan algoritme yang sama dengan yang terdapat pada perancangan komponen yaitu algoritme untuk mengambil data, algoritme mengakses API Twitter, dan algoritme untuk menentukan kategori sentimen.

1.4.1 Kode Program Mengambil Data

Nama Kelas : DataController

Nama Operasi : getSearchCityTwit(Request \$req)

Deskripsi : Kode Program ini digunakan ketika pengguna melakukan Pencarian data ujaran negatif yang dapat dilakukan dengan cara mengklik lokasi kota pada maps, memasukkan nama kota melalui pencarian kota, ataupun berdasarkan kota yang telah tersimpan didalam database. Algoritma ini berisi pengambilan data dari Twitter api dan melakukan klasifikasi dengan sentimen analisis untuk nantinya disimpan ke dalam db dan digunakan ke dalam *dashboard*

Kode Program :

Tabel 0.3 Kode Program Mengambil Data

| Kode program 1 : Fungsi mengambil data | |
|----------------------------------------|--------------------------------------------------------------|
| 1 | public function getSearchCityTwit(Request \$req){ |
| 2 | \$access_token = "3018593803-ue7gRf4vrd19gaAcTQbnzoDM2 |
| 3 | 14xcCsoAicCvrh"; |
| 4 | \$access_token_secret = |
| 5 | "MMHrkiFqwGVlEUCePUMOMElmkc3bBwxeG61OBgpjTuBu5"; |
| 6 | \$connection = new TwitterOAuth("t0ENtAtPJmuermYZPyW |
| 7 | bLfsZD", "nbuiH9TiZVTfvJGoL4DsquiztrDawgg1YI7an30am |
| 8 | OK6VE6NTQT", \$access_token, \$access_token_secret); |
| 9 | \$content = \$connection->get("account/verify_credentials"); |
| 10 | \$statuses = \$connection->get("search/tweets", |
| 11 | ["geocode"=>"\$req->lat,\$req->long,10mi","count"=>100]); |
| 12 | \$data = json_decode(json_encode(\$statuses),true); |
| 13 | \$countpos = 0; |
| 14 | \$countneg = 0; |
| 15 | \$countneu=0; |
| 16 | for (\$i=0; \$i <count(\$data["statuses"]); \$i++) { |
| 17 | \$sentiment = new SentimentAnalysis(); |
| 18 | \$nilai[] =\$sentiment->decision(\$data["statuses"] |
| 19 | ["\$i"]["text"]); |
| 20 | if(\$nilai[\$i]=="negative") { |
| 21 | \$countneg++; |
| 22 | } else if (\$nilai[\$i]=="positive") { |
| 23 | \$countpos++; |
| 24 | } else{ |

| Kode program 1 : Fungsi mengambil data | |
|----------------------------------------|----------------------------------------------------------------------|
| 25 | \$countneu++;}} |
| 26 | \$nama_kotas = DB::table('data_kota')->select('id', |
| 27 | 'nama_kota as kota')->get(); |
| 28 | \$kota = 0; |
| 29 | foreach (\$nama_kotas as \$nama_kota) { |
| 30 | if(strcasecmp(\$req->nama_kota, \$nama_kota->kota) == 0){ |
| 31 | \$kota++;}} |
| 32 | if(\$kota==0){ |
| 33 | DB::table('data_kota')->insert(['nama_kota' |
| 34 | =>\$req->nama_kota]); } |
| 35 | \$id = DB::table('data_kota')->where('nama_kota' |
| 36 | ,'=', \$req->nama_kota) |
| 37 | ->select('id') |
| 38 | ->first(); |
| 39 | \$ids = \$id->id; |
| 40 | DB::table('data_jam')->insert(['pos'=>\$countpos, 'neu'=>\$countneu, |
| 41 | 'neg'=>\$countneg, 'id_kota'=>\$ids]); |
| 42 | \$total = array("Kalimat Positif"=>\$countpos, "Kalimat |
| 43 | Negatif"=>\$countneg, "Kalimat Netral"=>\$countneu); |
| 44 | return json_encode(\$total); } |

1.4.2 Kode Program Mencari Persentase Ujaran Negatif

Nama Kelas : DataController

Nama Operasi : getPercentageNeg(Request \$req)

Deskripsi : Kode Program ini digunakan ketika pengguna mengklik *marker* untuk mendapatkan informasi mengenai persentase ujaran negatif didalam suatu kota. Didalamnya terdapat pengambilan data dari suatu kota dan perhitungan jumlah persentase untuk ujaran negatif dari kota yang dituju

Kode Program :

Tabel 0.4 Kode Program Mencari Persentase Ujaran Negatif

| Kode program 2 : Fungsi mencari persentase ujaran negatif | |
|-----------------------------------------------------------|----------------------------------------------------------------|
| 1 | public function getPercentageNeg(Request \$req){ |
| 2 | \$id = DB::table('data_kota')->where('nama_kota', '=', \$req-> |
| 3 | >nama_kota) |
| 4 | ->select('id') |
| 5 | ->first(); |
| 6 | if(isset(\$id)){ |
| 7 | \$ids = \$id->id; |
| 8 | \$neg=DB::table('data_jam')->where('id_kota', '=', \$ids)- |
| 9 | >select(DB::raw('SUM(neg) as negative'))->first(); |
| 10 | \$pos=DB::table('data_jam')->where('id_kota', '=', \$ids)- |
| 11 | >select(DB::raw('SUM(pos) as positive'))->first(); |
| 12 | \$neu=DB::table('data_jam')->where('id_kota', '=', \$ids)- |
| 13 | >select(DB::raw('SUM(neu) as neutral'))->first(); |
| 14 | \$Negatif = (int)\$neg->negative; |
| 15 | \$Positif = (int)\$pos->positive; |
| 16 | \$Neutral = (int)\$neu->neutral; |
| 17 | \$total=\$Negatif+\$Positif+\$Neutral; |
| 18 | \$percent = (\$Negatif/\$total)*100; |
| 19 | \$hasil = number_format(\$percent,2); |
| 20 | return json_encode(\$hasil); } |
| 21 | else{ |

| | |
|-----------------------------------------------------------|-----------------------------|
| Kode program 2 : Fungsi mencari persentase ujaran negatif | |
| 22 | return json_encode(null); } |
| 23 | |

1.4.3 Kode Program Mengambil Data 1 Hari Lalu

Nama Kelas : DataController

Nama Operasi : getTwitYesterday (Request \$req)

Deskripsi : Kode Program ini digunakan ketika pengguna mengklik lihat data 1 hari lalu pada *dashboard*. Didalam algoritme ini terdapat pengambilan data dengan query yang menyeleksi data berdasarkan waktu data diambil, yakni 1 hari sebelum pengguna menggunakan fungsi ini

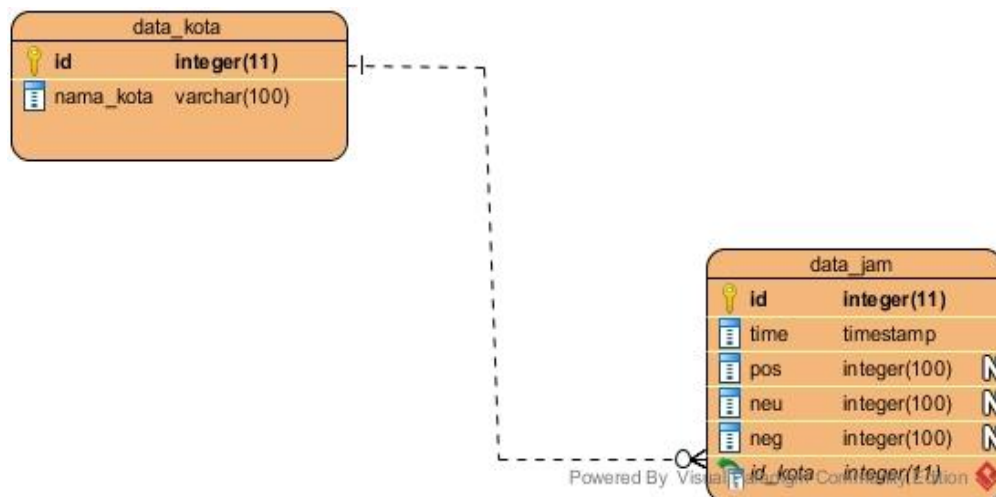
Kode Program :

Tabel 0.5 Kode Program Mengambil Data 1 Hari Lalu

| | |
|----------------------------------------------------|------------------------------------------------------------------------|
| Kode program 3 : Fungsi mengambil data 1 hari lalu | |
| 1 | public function getTwitYesterday(Request \$req){ |
| 2 | \$id = DB::table('data_kota')->where('nama_kota','=',\$req->nama_kota) |
| 3 | ->select('id') |
| 4 | ->first(); |
| 5 | \$ids = \$id->id; |
| 6 | \$data=DB::table('data_jam')->where([|
| 7 | ['id_kota','=',\$ids], |
| 8 | ['time','>',Carbon::yesterday(), |
| 9 | ['time','<',Carbon::today()], |
| 10 |]) |
| 11 | ->select('neg','time') |
| 12 | ->get(); |
| 13 | if(isset(\$data)){ |
| 14 | return json_encode(\$data); |
| 15 | } |
| 16 | else{ |
| 17 | return json_encode(null); |
| 18 | } |
| 19 | } |

1.5 Implementasi Basis Data

Implementasi basis data pada sistem monitoring *cyberbullying* ini didasarkan pada perancangan basis data pada bab sebelumnya. Pada tahapan implementasi basis data ini ditunjukkan dalam bentuk *Physical Data Model* (PDM). *Physical Data Model* merupakan tahapan untuk mentransformasikan hasil identifikasi entitas yang telah dilakukan pada *Conceptual Data Model* ke dalam tabel database fisik yang terperinci. Hasil Implementasi basis data pada sistem *dashboard* monitoring *cyberbullying* dapat dilihat pada Gambar 6.1 berikut.



Gambar 0.1 Physical Data Model Sistem Dashboard Monitoring Cyberbullying

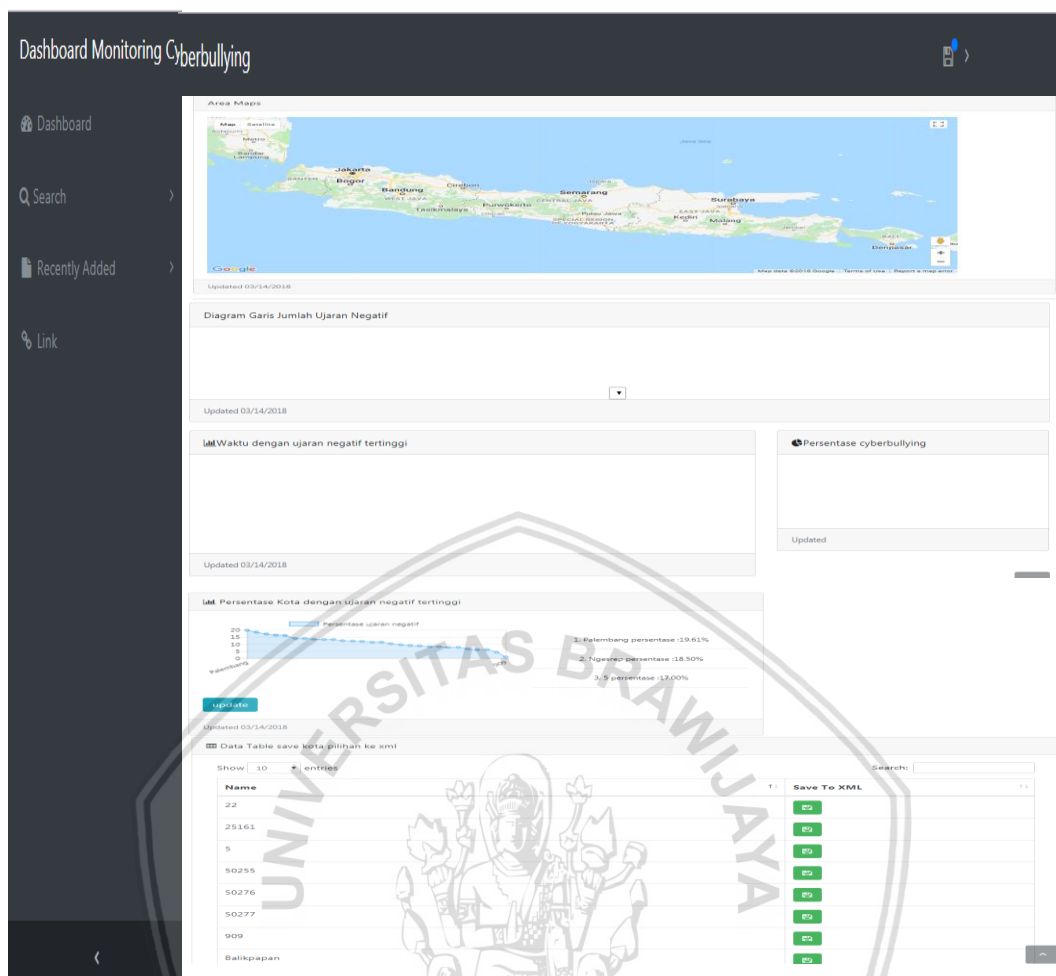
Pada implementasi basis data diatas menunjukkan tabel-tabel yang nantinya menyusun sistem yang dibangun, dimana terdapat dua buah tabel yaitu tabel data_kota dan data_jam. Tabel data_kota merupakan tabel yang digunakan untuk menyimpan kota-kota yang nantinya dimonitoring, terdiri dari dua kolom yakni id dan nama_kota. Tabel data_jam merupakan tabel yang digunakan untuk menyimpan data hasil klasifikasi tweet dari kota yang dimonitoring, didalamnya terdapat kolom id, time, pos, neg, neu, id_kota. Kedua tabel tersebut berhubungan one to many dengan tabel data_kota pada kolom id sebagai primary key yang merujuk kepada foreign key tabel data_jam pada kolom id_kota.

1.6 Implementasi Antar Muka

Implementasi antar muka merupakan proses pembuatan antarmuka yang dilakukan berdasarkan perancangan antar muka. Berikut adalah tampilan implementasi antar muka *dashboard* sistem monitoring persebaran *cyberbullying*.

1.6.1 Halaman Home

Halaman home merupakan halaman utama dalam *dashboard* sistem monitoring *cyberbullying*. Pada halaman ini terdapat seluruh fungsionalitas yang dapat dijalankan oleh pengguna.



Gambar 0.2 Implementasi Antar Muka Halaman Home

Gambar 6.2 menunjukkan halaman utama dari *dashboard* sistem monitoring *cyberbullying*. Semua fungsionalitas disediakan hanya pada satu halaman ini. Pengguna dapat memilih kota yang akan diamati persebaran *cyberbullying* nya dengan melakukan klik di maps yang tersedia, memasukkan nama kota pada kolom pencarian, serta kota yang telah tersimpan sebelumnya. Persebaran ditampilkan dengan diagram garis yang menampilkan jumlah ujaran negatif pada kota yang di klik *markernya*. Pengguna dapat melihat waktu persebaran dengan jumlah ujaran negatif tertinggi pada kota yang dipilih serta dapat melihat persentase ujaran negatif pada kota tersebut tersebut. Pada bagian persentase kota dengan ujaran negatif tertinggi pengguna dapat melihat peringkat kota dengan ujaran negatif tertinggi dengan diagram garis serta pengguna dapat melihat tiga kota dengan urutan tertinggi. Pada bagian yang berbentuk data tabel pengguna dapat melakukan *export* data ke dalam bentuk excel berdasarkan kota yang dipilih oleh pengguna dalam tabel tersebut. Pengguna dapat melakukan *export* seluruh data kota yang tersimpan dengan mengklik *icon save* pada bagian navbar.

BAB 7 PENGUJIAN

Pengujian merupakan sebuah proses untuk memastikan fungsionalitas yang telah diimplementasikan dapat berjalan sesuai dengan tujuan fungsionalitas tersebut, dan untuk memastikan fungsionalitas bebas dari adanya kegagalan. Pengujian dalam pengembangan sistem ini dilakukan dengan pengujian unit, pengujian integrasi, dan pengujian validasi.

1.1 Pengujian Unit

Pengujian unit merupakan tahapan pengujian yang digunakan untuk menguji komponen individu atau sebuah subsistem dari sebuah class yang bertujuan untuk memastikan komponen yang diimplementasikan telah berjalan sesuai dengan tujuan dari fungsionalitas. Pada pengujian unit ini menggunakan metode pengujian *white box* dengan jenis pengujian basis path. Pada pengujian unit ini dilakukan kepada tiga sampel uji yaitu pada tiga operasi yang ada pada kelas `data_controller`.

1.1.1 Pengujian Unit Kelas `data_controller` untuk operasi `getSearchCityTwit(Request)`

1. Pseudocode

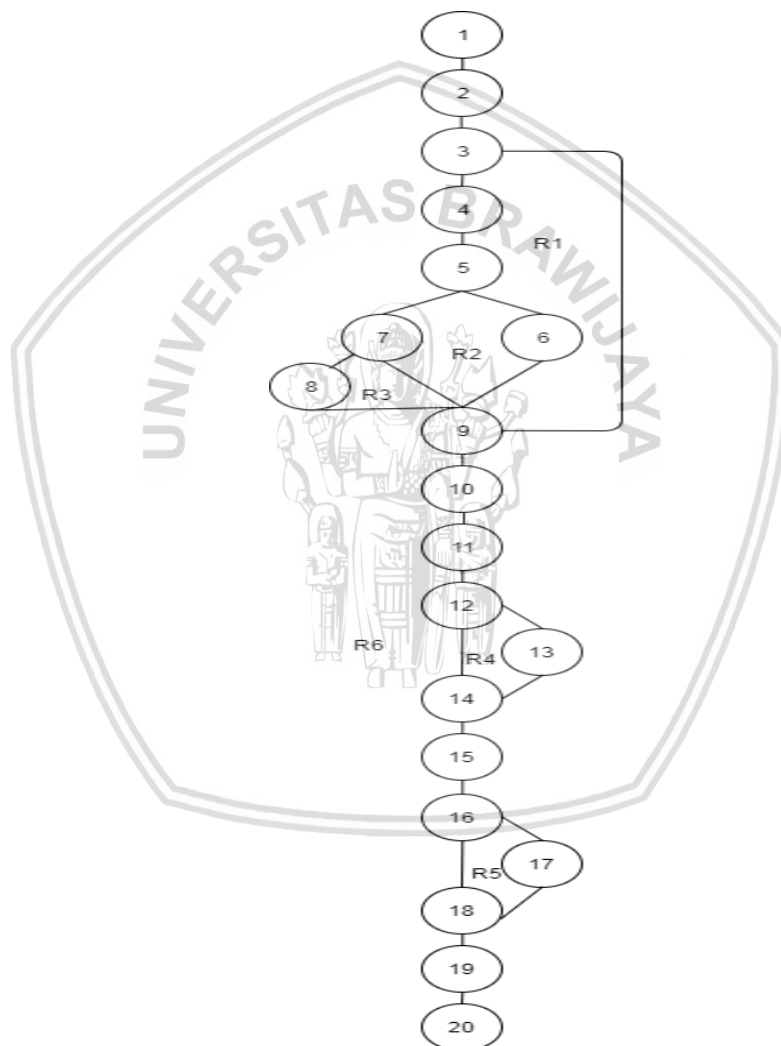
Tabel 0.1 Pseudocode Pengujian Unit Operasi `getSearchCityTwit(Request)`

| Pseudocode | No. operasi |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| inisialisasi access token, access token secret instansiasi class <code>TwitterOAuth</code> inisialisasi content untuk memanggil method <code>get</code> untuk verify akun dan mengambil data dari endpoint API Twitter inisialisasi statuses untuk mengambil data dengan method <code>get</code> dari kelas <code>TwitterOAuth</code> inisialisasi data = decode statuses | 1 |
| inisialisasi <code>countpos</code> , <code>countneg</code> , <code>countneu</code> = 0 | 2 |
| loop for dari <code>i</code> = 0 sampai jumlah data | 3 |
| instansiasi kelas <code>SentimentAnalysis</code> inisialisasi array nilai = dilakukan penentuan positif, negatif atau netral dengan memanggil method <code>desicion</code> pada kelas <code>SentimentAnalysis</code> pada tweet ke- <code>i</code> | 4 |
| seleksi if jika nilai ke- <code>i</code> = negatif | 5 |
| <code>countneg++</code> | 6 |
| else if nilai ke- <code>i</code> = positif <code>countpos++</code> | 7 |
| else <code>countneu++</code> | 8 |
| endif | 9 |
| inisialisasi nama kotas yaitu mengambil data dari tabel data kota dengan mengambil id, dan nama kota inisialisasi kota = 0 | 10 |
| loop foreach nama kotas as nama kota | 11 |
| if nama kota pada request == nama kota | 12 |
| <code>kota++</code> | 13 |
| endif | 14 |
| endforeach | 15 |
| if kota=0 | 16 |

| Pseudocode | No. operasi |
|----------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| insert kota ke tabel data kota | 17 |
| endif | 18 |
| inisialisasi id = mengambil id berdasarkan nama kota request dari tabel data_kota insert countpos,countneg,countneu kedalam data_jam bersama id | 19 |
| inisialisasi total return total | 20 |

2. Basic path testing

a. Flow Graph



Gambar 0.1 Flowgraph Pengujian Unit Operasi getSearchCityTwit(Request)

b. Cyclomatic Complexity

$V(G) = 6$, ada 6 region R1, R2, R3, R4, R5, R6

$V(G) = 24 \text{ edge} - 20 \text{ node} + 2 = 6$

$V(G) = 5 \text{ predicate node} + 1 = 6$

c. Independen Path

Jalur 1 = 1-2-3-4-5-7-8-9-10-11-12-14-15-16-17-18-19-20

Jalur 2 = 1-2-3-4-5-7-9-10-11-12-14-15-16-17-18-19-20

Jalur 3 = 1-2-3-4-5-6-9-10-11-12-14-15-16-17-18-19-20

Jalur 4 = 1-2-3-4-5-7-8-9-10-11-12-13-14-15-16-18-19-20

Jalur 5 = 1-2-3-4-5-7-9-10-11-12-13-14-15-16-18-19-20

Jalur 6 = 1-2-3-4-5-6-9-10-11-12-13-14-15-16-18-19-20

Tabel 0.2 Pengujian Unit Kelas `data_controller` untuk operasi `getSearchCityTwit` (Request)

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|--------|
| 1 | 1 | 1. Variabel \$nilai pada looping ke i bernilai = "netral" 2. Variabel \$kota bernilai 0 | Akan menambahkan data netral ke dalam db <i>dashboard</i> dan menambahkan nama kota pada tabel <i>data_kota</i> pada database | Menambahkan data netral ke dalam db <i>dashboard</i> dan menambahkan nama kota pada tabel <i>data_kota</i> pada database | valid |
| 2 | 2 | 1. Variabel \$nilai pada looping ke i bernilai = "positif" 2. Variabel \$kota bernilai 0 | Akan menambahkan data positif ke dalam db <i>dashboard</i> dan menambahkan nama kota pada tabel <i>data_kota</i> pada database | Menambahkan data positif ke dalam db <i>dashboard</i> dan menambahkan nama kota pada tabel <i>data_kota</i> pada database | valid |
| 3 | 3 | 1. Variabel \$nilai pada looping ke i bernilai = "negatif" 2. Variabel \$kota bernilai 0 | Akan menambahkan data negatif ke dalam db <i>dashboard</i> dan menambahkan nama kota pada tabel <i>data_kota</i> pada database | Menambahkan data negatif ke dalam db <i>dashboard</i> dan menambahkan nama kota pada tabel <i>data_kota</i> pada database | valid |
| 4 | 4 | 1. Variabel \$nilai pada looping | Akan menambahkan data netral ke | Menambahkan data netral ke dalam db | valid |

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|--------|
| | | ke i bernilai = "netral" 2. Variabel \$kota bernilai != 0 | dalam db <i>dashboard</i> tanpa menambah kan nama kota pada tabel data_kota pada database | <i>dashboard</i> tanpa menambah kan nama kota pada tabel data_kota pada database | |
| 5 | 5 | 1. Variabel \$nilai pada looping ke i bernilai = "positif" 2. Variabel \$kota bernilai != 0 | Akan menambahkan data positif ke dalam db <i>dashboard</i> tanpa menambah kan nama kota pada tabel data_kota pada database | Menambahkan data positif ke dalam db <i>dashboard</i> tanpa menambah kan nama kota pada tabel data_kota pada database | valid |
| 6 | 6 | 1. Variabel \$nilai pada looping ke i bernilai = "negatif" 2. Variabel \$kota bernilai != 0 | Akan menambahkan data negatif ke dalam db <i>dashboard</i> tanpa menambah kan nama kota pada tabel data_kota pada database | Menambahkan data negatif ke dalam db <i>dashboard</i> tanpa menambah kan nama kota pada tabel data_kota pada database | valid |

1.1.2 Pengujian Unit Kelas data_controller untuk operasi getPercentageNeg(Request)

1. Pseudocode

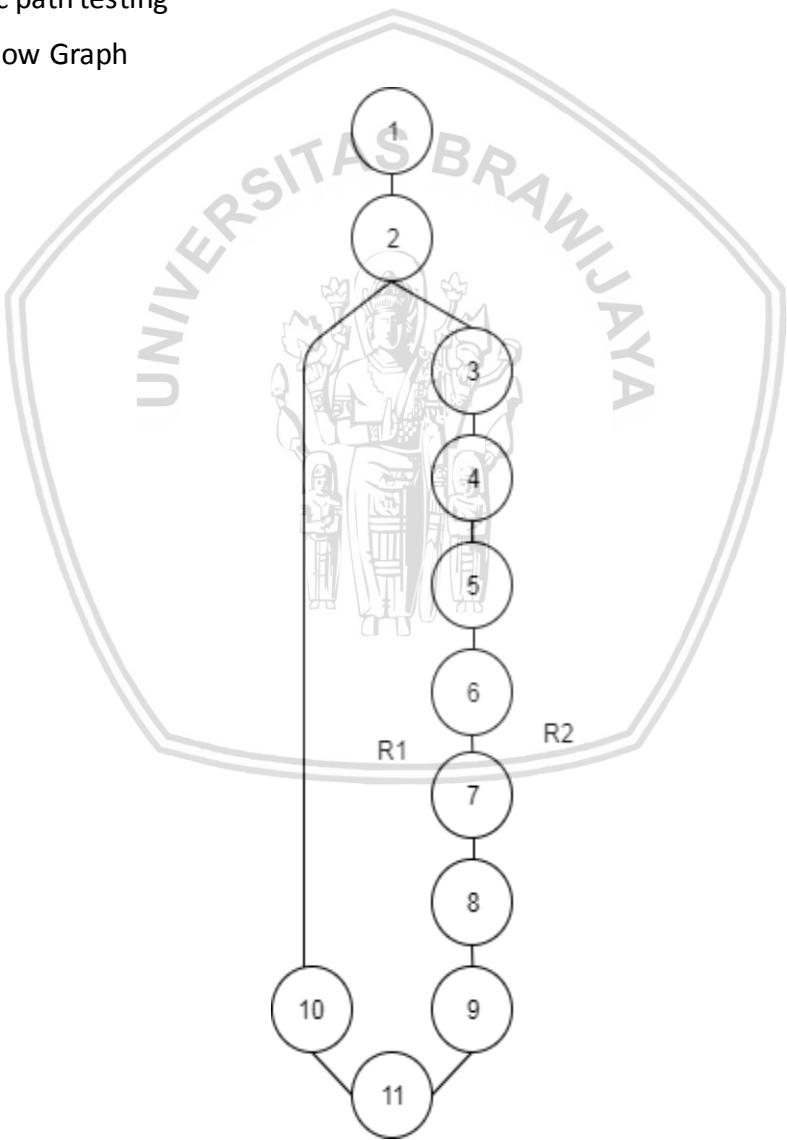
Tabel 0.3 Pseudocode Pengujian Unit Operasi getPercentageNeg(Request)

| Pseudocode | No. operasi |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|
| inisialisasi variable id = select id dari table data kota berdasarkan nama kota yang diterima controller | 1 |
| percabangan if jika variabel id terdapat nilainya | 2 |
| var ids = id | 3 |
| inisialisasi variabel neg = select jumlah neg tabel data_jam dengan id kota = ids inisialisasi variabel pos = select jumlah pos tabel data_jam dengan id kota = ids inisialisasi variabel neu = select jumlah neu tabel data_jam dengan id kota = ids | 4 |
| inisialisasi variabel Negatif = hasil query variable neg | 5 |

| Pseudocode | No. operasi |
|-----------------------------------------------------------------------------|-------------|
| inisialisasi variabel Positif = hasil query variable pos | |
| inisialisasi variabel Neutral = hasil query variable neu | |
| inisialisasi variabel total = Positif+Negatif+neutral | 6 |
| inisialisasi variable percent = (Negatif/total)*100; | 7 |
| inisialisasi hasil = format dua angka dibalekang koma dari variable percent | 8 |
| return json_encode(hasil) | 9 |
| endif | |
| else | |
| return json_encode(null) | 10 |
| end ifelse | |
| end | 11 |

2. Basic path testing

a. Flow Graph



Gambar 0.2 Flowgraph Pengujian Unit Operasi getPercentageNeg(Request)

b. Cyclomatic Complexity

$$V(G) = 2, \text{ ada 2 region } R1, R2$$

$$V(G) = 11 \text{ edge} - 11 \text{ node} + 2 = 2$$

$$V(G) = 1 \text{ predicate node} + 1 = 2$$

c. Independen Path

Jalur 1 = 1-2-3-4-5-6-7-8-9-10-11

Jalur 2 = 1-10-11

Tabel 0.4 Pengujian Unit Kelas data_controller untuk operasi getPercentageNeg (Request)

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|-----------------------------------|------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|--------|
| 1 | 1 | 1. Variabel \$id bernilai != null | Akan Menampilkan info content pada marker dengan hasil persentase ujaran negatif pada lokasi marker yang di klik | Menampilkan info content pada marker dengan hasil persentase ujaran negatif pada lokasi marker yang di klik | valid |
| 2 | 2 | 1. Variabel \$id bernilai = null | Akan Menampilkan info content pada marker tanpa hasil persentase ujaran negatif pada kota yang di klik | Menampilkan info content pada marker tanpa hasil persentase ujaran negatif pada kota yang di klik | valid |

1.1.3 Pengujian Unit Kelas data_controller untuk operasi getTwitYesterday(Request)

1. Pseudocode

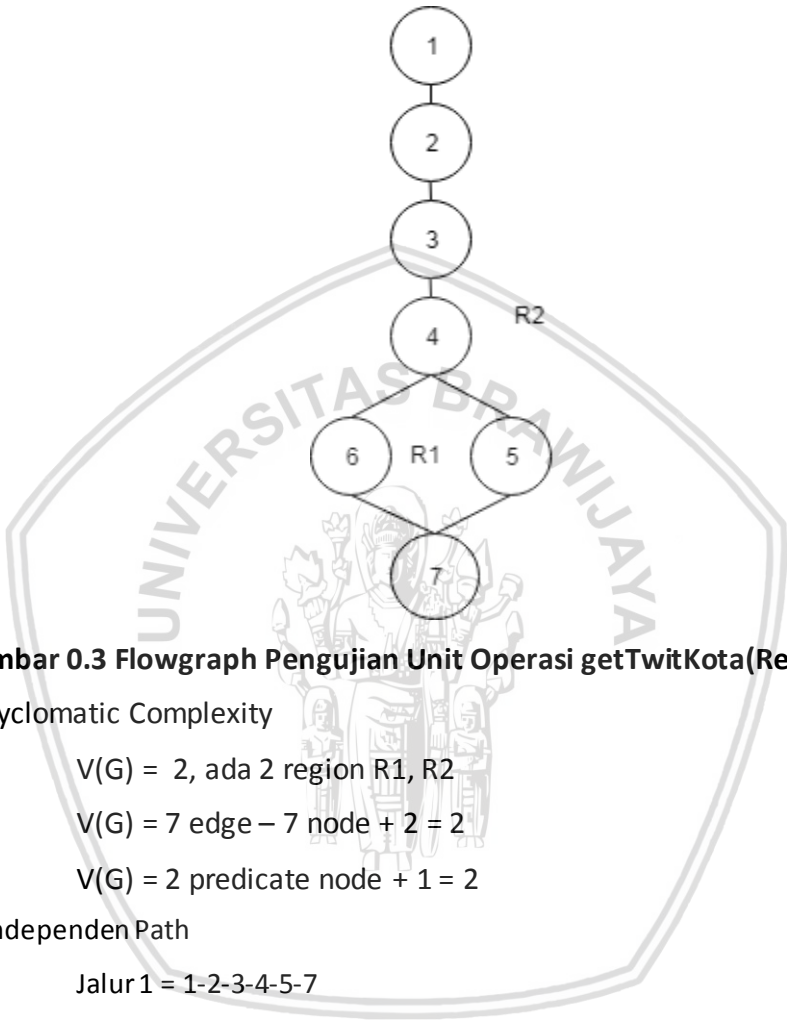
Tabel 0.5 Pseudocode Pengujian Unit Operasi getTwitYesterday(Request)

| Pseudocode | No. operasi |
|----------------------------------------------------------------------------------------------------------|-------------|
| inisialisasi variable id = select id dari table data kota berdasarkan nama kota yang diterima controller | 1 |
| inisialisasi variable ids = hasil query dr id | 2 |
| inisialisasi variable data = select neg, dan time berdasarkan ids time > yesterday, time < today | 3 |
| percabangan if jika variable data diset | 4 |
| return json_encode(data); | 5 |

| | |
|---------------------------|---|
| else | |
| return json_encode(null); | 6 |
| end if | 7 |
| end | |

2. Basic path testing

a. Flow Graph



Gambar 0.3 Flowgraph Pengujian Unit Operasi getTwitKota(Request)

b. Cyclomatic Complexity

$$V(G) = 2, \text{ ada 2 region R1, R2}$$

$$V(G) = 7 \text{ edge} - 7 \text{ node} + 2 = 2$$

$$V(G) = 2 \text{ predicate node} + 1 = 2$$

c. Independen Path

$$\text{Jalur 1} = 1-2-3-4-5-7$$

$$\text{Jalur 1} = 1-2-3-4-6-7$$

Tabel 0.6 Pengujian Unit Kelas data_controller untuk operasi getTwitYesterday(Request)

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|-------------------------------------|-------------------------------------------------------------------------|--------------------------------------------------------------------|--------|
| 1 | 1 | 1. Variabel \$data bernilai != null | Akan menampilkan diagram garis dengan data 1 hari lalu sebelum pengguna | Menampilkan diagram garis dengan data 1 hari lalu sebelum pengguna | valid |

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|------------------------------------|--------------------------------------------------------|---------------------------------------------------|--------|
| | | | menggunakan sistem | menggunakan sistem | |
| 2 | 2 | 1. Variabel \$data bernilai = null | Akan menampilkan diagram garis dengan data yang kosong | Menampilkan diagram garis dengan data yang kosong | valid |

1.2 Pengujian Integrasi

Pengujian integrasi merupakan merupakan jenis pengujian yang menguji kelas-kelas yang saling berhubungan. Pengujian ini berfokus pada hubungan dan interaksi antar kelas yang dibangun berdasarkan arsitektur kelas yang telah didefinisikan pada perancangan. Pengujian integrasi dilakukan dengan teknik *white box testing* dengan menggunakan *basis path testing*. Pengujian ini akan dilakukan pada operasi *decision(\$string)* pada kelas *SentimentAnalysis* yang berhubungan dengan kelas *twitController* pada operasi *getTwit()* dan operasi *score(\$sentence)* pada kelas *Sentiment* yang berhubungan dengan operasi *decision(\$string)* pada kelas *SentimentAnalysis*.

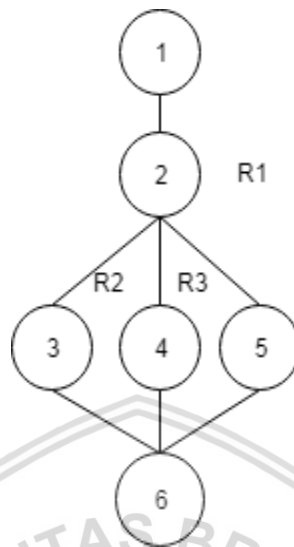
1.2.1 Operasi decision(String)

1. Pseudocode

Tabel 0.7 Pseudocode Operasi decision(String)

| Pseudocode | No. operasi |
|-------------------------------------------------------------------------|-------------|
| <code>dominantClass = mengambil klasifikasi berdasarkan kalimat</code> | 1 |
| <code>percabangan switch(dominantClass)</code> | 2 |
| <code>ketika dominantClass = 'neg': mengembalikan "negatif";</code> | 3 |
| <code>ketika dominantClass = 'neu': mengembalikan "netral";</code> | 4 |
| <code>ketika dominantClass = 'pos': mengembalikan "positif";</code> | 5 |
| <code>end percabangan switch</code> | 6 |

2. Basis Path testing
 - a. Flow Graph



Gambar 0.4 Flowgraph Pengujian Integrasi Operasi decision(String)

- b. Cyclomatic Complexity
 $V(G) = 3$ ada 3 region R1, R2, R3
- c. Independen Path
Jalur 1 : 1-2-5-6
Jalur 2 : 1-2-4-6
Jalur 3 : 1-2-3-6

Tabel 0.8 Pengujian Integrasi Operasi decision(String)

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|----------------------------------------------|------------------------------------------------------------------|-------------------------------------------------------------|--------|
| 1 | 1 | 1. Variabel \$dominantClass bernilai = 'neg' | Akan mengembalikan hasil negatif pada pemanggilan kelas tersebut | Mengembalikan hasil negatif pada pemanggilan kelas tersebut | valid |
| 2 | 2 | 1. Variabel \$dominantClass bernilai = 'neu' | Akan mengembalikan hasil netral pada pemanggilan kelas tersebut | Mengembalikan hasil netral pada pemanggilan kelas tersebut | valid |

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|----------------------------------------------------|---------------------------------------------------------------------------------|-------------------------------------------------------------------------|--------|
| 3 | 3 | 1. Variabel \$dominantClass bernilai = 'pos' | Akan mengembalikan hasil positif pada pemanggilan kelas tersebut | Mengembalikan hasil positif pada pemanggilan kelas tersebut | valid |

1.2.2 Operasi score()

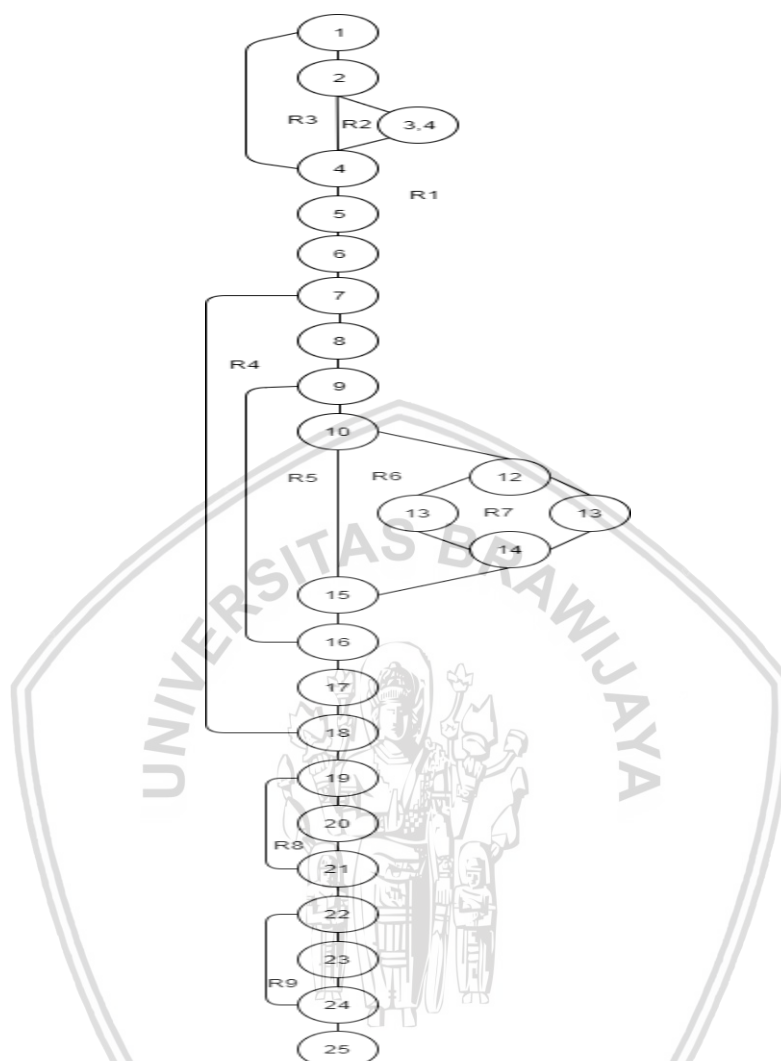
1. Pseudocode

Tabel 0.9 Pseudocode Operasi score()

| Pseudocode | No. operasi |
|-----------------------------------------------------------------------------------------------------------------------------|-------------|
| Looping foreach negativePrefix | 1 |
| Seleksi if jika sentence yang dikirimkan mengandung prefix | 2 |
| Menghapus spasi setelah prefix pada sentence | 3 |
| endif | 4 |
| endforeach | 5 |
| inisialisasi \$tokens = getTokens(\$sentence); inisialisasi \$total_score = 0; inisialisasi array \$scores = array(); | 6 |
| looping foreach seluruh classes | 7 |
| mengisi array score dengan dimensi class dengan value = 1 | 8 |
| looping tiap kata pada sentence | 9 |
| percabangan if jika kata melebihi syarat tokenize | 10 |
| percabangan if jika kata terdapat pada class dalam dictionary | 11 |
| inisialisasi count = nilai dari \$this->dictionary[\$token][\$class] | 12 |
| } else { Inisialisasi count = 0 | 13 |
| } Inisialisasi scores pada class dengan scores*(count+1) | 14 |
| End if | 15 |
| End foreach | 16 |
| inisialisasi scores pada class dengan prior[class]*scores[class] | 17 |
| endforeach | 18 |
| looping foreach class | 19 |
| inisialisasi total score = total_score+score[class] | 20 |
| endforeach | 21 |
| looping foreach class | 22 |
| inisialisasi score[class] = score[class]/total_score | 23 |
| endforeach | 24 |
| sorting array score[class] return scores; | 25 |

2. Basis Path Testing

a. Flow Graph



Gambar 0.5 Flowgraph Operasi score()

b. Cyclomatic Complexity

$V(G) = 9$ ada 9 region R1, R2, R3, R4, R5, R6, R7, R8, R9

$V(G) = 32 \text{ edge} - 25 \text{ node} + 2 = 9$

$V(G) = 8 \text{ predicate node} + 1 = 9$

c. Independen Path

Jalur 1 = 1-5-6-7-18-19-21-22-24-25

Jalur 2 = 1-5-6-7-8-9-16-17-18-19-20-21-22-23-24-25

Jalur 3 = 1-2-5-6-7-8-9-10-15-16-17-18-19-20-21-22-23-24-25

Jalur 4 = 1-2-5-6-7-8-9-10-11-13-14-15-16-17-18-19-20-21-22-23-24-25

Jalur 5 = 1-2-5-6-7-8-9-10-11-12-14-15-16-17-18-19-20-21-22-23-24-25

Jalur 6 = 1-2-3-4-5-6-7-8-9-10-11-13-14-15-16-17-18-19-20-21-22-23-24-25

Jalur 7 = 1-2-3-4-5-6-7-8-9-10-11-12-14-15-16-17-18-19-20-21-22-23-24-25

Jalur 8 = 1-2-3-4-5-6-7-8-9-10-15-16-17-18-19-20-21-22-23-24-25

Jalur 9 = 1-5-6-7-8-9-10-15-16-17-18-19-20-21-22-23-24-25

Tabel 0.10 Pengujian Integrasi Operasi score()

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| 1 | 1 | 1. Variabel \$class bernilai = 'neu' | Akan mengembalikan array scores dengan urutan Masuk ke dalam looping terakhir foreach negativePrefix dengan looping terakhir class serta looping terakhir pada inialisasi total_score dan score[class] dan mengembalikan score | Mengembalikan array scores dengan urutan Masuk ke dalam looping terakhir foreach negativePrefix dengan looping terakhir class serta looping terakhir pada inialisasi total_score dan score[class] dan mengembalikan score | Valid |
| 2 | 2 | 1. Variabel \$class bernilai = 'neu' 2. Variabel \$token bernilai != \$this->ignorelist | Akan mengembalikan array scores dengan urutan Masuk Looping terakhir foreach negativePrefix dengan looping class dan mengisi array score dengan value = 1 serta looping terakhir kata pada | Mengembalikan array scores dengan urutan masuk Looping terakhir foreach negativePrefix dengan looping class dan mengisi array score dengan value = 1 serta looping terakhir kata pada sentence serta | Valid |

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|--------|
| | | | sentence serta pada inialisasi total_score , score[class] | pada inialisasi total_score , score[class] | |
| 3 | 3 | 1. Variabel \$sentence tidak memiliki prefix 2. Variabel \$token bernilai != \$this->ignorelist | Akan mengembalikan array scores dengan Sentence tidak mengandung prefix, kata tidak memenuhi tokenize | Mengembalikan array scores dengan Sentence tidak mengandung prefix, kata tidak memenuhi tokenize | valid |
| 4 | 4 | 1. Variabel \$sentence tidak memiliki prefix 2. Variabel \$token bernilai = \$this->ignorelist 3. Variabel \$this->dictionary [\$token] [\$class] bernilai != null | Akan mengembalikan array scores dengan Sentence tidak mengandung prefix, kata memenuhi syarat tokenize, inialisasi count = 0 | Akan mengembalikan array scores dengan Sentence tidak mengandung prefix, kata memenuhi syarat tokenize, inialisasi count = 0 | Valid |
| 5 | 5 | 1. Variabel \$sentence tidak memiliki prefix 2. Variabel \$token bernilai == \$this->ignorelist | Akan mengembalikan array scores dengan Sentence tidak mengandung prefix, kata memenuhi syarat tokenize, inialisasi count | Akan mengembalikan array scores dengan Sentence tidak mengandung prefix, kata memenuhi syarat tokenize, inialisasi count | Valid |

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| | | 3. Variabel \$this->dictionary [\$token] [\$class] bernilai = null | = nilai dari dictionary berdasarkan token dan class | = nilai dari dictionary berdasarkan token dan class | |
| 6 | 6 | 1. Variabel \$sentence memiliki prefix 2. Variabel \$token bernilai = \$this->ignorelist 3. Variabel \$this->dictionary [\$token] [\$class] bernilai != null | Akan mengembalikan array scores dengan Sentence mengandung prefix, kata memenuhi syarat tokenize, inialisasi count = 0 | Akan mengembalikan array scores dengan Sentence mengandung prefix, kata memenuhi syarat tokenize, inialisasi count = 0 | Valid |
| 7 | 7 | 1. Variabel \$sentence memiliki prefix 2. Variabel \$token bernilai = \$this->ignorelist 3. Variabel \$this->dictionary [\$token] [\$class] bernilai null | Akan mengembalikan array scores dengan Sentence mengandung prefix, kata memenuhi syarat tokenize, inialisasi count = nilai dari dictionary berdasarkan token dan class | Akan mengembalikan array scores dengan Sentence mengandung prefix, kata memenuhi syarat tokenize, inialisasi count = nilai dari dictionary berdasarkan token dan class | Valid |
| 8 | 8 | 1. Variabel \$sentence memiliki prefix | Akan mengembalikan array scores dengan Sentence | Mengembalikan array scores dengan Sentence mengandung | Valid |

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|---------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|
| | | 2. Variabel \$token bernilai != \$this->ignorelist | mengandung prefix, kata tidak memenuhi syarat tokenize dengan sebelumnya melakukan proses pengecekan negatif prefix | prefix, kata tidak memenuhi syarat tokenize dengan sebelumnya melakukan proses pengecekan negatif prefix | |
| 9 | 9 | 1. Looping terakhir \$negPrefix 2. Variabel \$sentence tidak memiliki prefix 3. Variabel \$token bernilai != \$this->ignorelist | Akan mengembalikan array scores dengan Sentence mengandung prefix, kata tidak memenuhi syarat tokenize dengan sebelumnya telah melakukan proses pengecekan negatif prefix | Akan mengembalikan array scores dengan Sentence mengandung prefix, kata tidak memenuhi syarat tokenize dengan sebelumnya telah melakukan proses pengecekan negatif prefix | Valid |

1.3 Pengujian Validasi

Pengujian validasi merupakan tahapan pengujian yang dilakukan untuk memeriksa kesesuaian antara fungsi-fungsi didalam sistem dengan kebutuhan yang didefinisikan sebelumnya didalam analisis kebutuhan. Pengujian validasi dilakukan dengan memastikan tidak adanya penyimpangan dari fungsi yang dibangun didalam sistem dan berjalan sesuai yang diharapkan. Pada pengujian ini dilakukan dengan black box testing yakni teknik pengujian tanpa perlu mengetahui kode program dan hanya menjalankan setiap fungsi dan menyesuaikan dengan kebutuhan yang telah didefinisikan.

1.3.1 Kebutuhan Fungsional

1.3.1.1 Pengujian Validasi Mengambil Data

Tabel 0.11 Kasus Uji Mengambil Data

| | |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | Pengujian mengambil data |
| Prosedur | <ol style="list-style-type: none"> 1. Pengguna telah membuka halaman home dan <i>browser</i> telah meload peta 2. Pengguna melakukan klik kiri pada lokasi yang ada di maps |
| Hasil Yang Diharapkan | Sistem akan memulai proses pengambilan data untuk setiap interval waktu tertentu pada lokasi yang telah ditentukan pengguna yang ditunjukkan dengan <i>marker</i> |
| Hasil | Sistem memulai proses pengambilan data untuk setiap interval waktu tertentu pada lokasi yang telah ditentukan pengguna yang ditunjukkan dengan <i>marker</i> |
| Status | Valid |

1.3.1.2 Pengujian Validasi Mengambil Data Dengan Form Pencarian

Tabel 0.12 Kasus Uji Mengambil Data Dengan Form Pencarian

| | |
|-----------------------|----------------------------------------------------------------------------------|
| Nama Kasus Uji | Pengujian mengambil data dengan form pencarian |
| Prosedur | <ol style="list-style-type: none"> 1. Mengklik menu <i>search</i> |

| | |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | <ol style="list-style-type: none"> Pengguna memasukkan nama kota pada textfield Pengguna menekan tombol yang memiliki <i>icon</i> search |
| Hasil Yang Diharapkan | Sistem akan memulai proses pengambilan data untuk setiap interval waktu tertentu pada lokasi yang diinputkan pengguna yang ditunjukkan dengan <i>marker</i> |
| Hasil | Sistem akan memulai proses pengambilan data untuk setiap interval waktu tertentu pada lokasi yang diinputkan pengguna yang ditunjukkan dengan <i>marker</i> |
| Status | Valid |

1.3.1.3 Pengujian Validasi Mengambil Data Pada Kota Yang Tersimpan

Tabel 0.13 Kasus Uji Mengambil Data Pada Kota Yang Tersimpan

| | |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | Pengujian Mengambil Data Pada Kota Yang Tersimpan |
| Prosedur | <ol style="list-style-type: none"> Pengguna mengklik menu <i>recently added</i> Pengguna memilih list kota yang tersedia Pengguna menekan tombol yang memiliki <i>icon</i> tambah |
| Hasil Yang Diharapkan | Sistem akan memulai proses pengambilan data untuk setiap interval waktu tertentu pada lokasi yang telah dipilih pengguna yang ditunjukkan dengan <i>marker</i> |
| Hasil | Sistem memulai proses pengambilan data untuk setiap interval waktu tertentu pada lokasi yang telah dipilih pengguna yang ditunjukkan dengan <i>marker</i> |
| Status | Valid |

1.3.1.4 Pengujian Validasi Melihat Persentase

Tabel 0.14 Kasus Uji Melihat Persentase

| | |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | Melihat persentase |
| Prosedur | <ol style="list-style-type: none"> 1. Pada <i>maps</i> telah terdapat <i>marker</i> pada kota yang dituju 2. Pengguna melakukan klik kiri pada <i>marker</i> |
| Hasil Yang Diharapkan | Pada bagian atas <i>marker</i> akan menampilkan info konten yang berisi persentase ujaran negatif, serta persentase ujaran negatif akan ditampilkan dengan diagram <i>pie</i> pada bagian persentase <i>cyberbullying</i> |
| Hasil | <i>marker</i> akan menampilkan info konten yang berisi persentase ujaran negatif, serta persentase ujaran negatif akan ditampilkan dengan diagram <i>pie</i> pada bagian persentase <i>cyberbullying</i> |
| Status | Valid |

1.3.1.5 Pengujian Validasi Melihat Detail Data

Tabel 0.15 Kasus Uji Melihat Detail Data

| | |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | Melihat seluruh detail data |
| Prosedur | <ol style="list-style-type: none"> 1. Pada <i>maps</i> telah terdapat <i>marker</i> pada kota yang dituju 2. Pengguna melakukan klik kiri pada <i>marker</i> atau menekan button lihat seluruh data |
| Hasil Yang Diharapkan | Pada bagian diagram garis jumlah ujaran negatif, akan menampilkan informasi mengenai seluruh data ujaran negatif beserta waktunya, yang telah tersimpan di dalam database serta |

| | |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| | ditunjukkan dengan menggunakan diagram garis |
| Hasil | Menampilkan informasi mengenai seluruh data ujaran negatif beserta waktunya, yang telah tersimpan di dalam database serta ditunjukkan dengan menggunakan diagram garis |
| Status | Valid |

1.3.1.6 Pengujian Validasi Melihat Detail Data Tanggal Pilihan

Tabel 0.16 Kasus Uji Melihat Detail Data Tanggal Pilihan

| | |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | Melihat detail data tanggal pilihan |
| Prosedur | <ol style="list-style-type: none"> 1. Pada maps telah terdapat <i>marker</i> pada kota yang dituju 2. Pengguna melakukan klik kiri pada <i>marker</i> 3. Pengguna memilih menu <i>dropdown</i> tanggal yang tersedia pada lokasi yang dipilih sebelumnya |
| Hasil Yang Diharapkan | Pada bagian diagram garis jumlah ujaran negatif, akan menampilkan informasi mengenai seluruh data ujaran negatif yang tersimpan di dalam database berdasarkan tanggal yang dipilih pengguna yang divisualisasikan dengan menggunakan diagram garis |
| Hasil | Menampilkan informasi mengenai seluruh data ujaran negatif yang tersimpan di dalam database berdasarkan tanggal yang dipilih pengguna yang divisualisasikan dengan menggunakan diagram garis |
| Status | Valid |

1.3.1.7 Pengujian Validasi Melihat Detail Data Hari Ini

Tabel 0.17 Kasus Uji Lihat Melihat Detail Data Hari Ini

| | |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | Melihat detail data hari ini |
| Prosedur | <ol style="list-style-type: none"> 1. Pada maps telah terdapat <i>marker</i> pada lokasi yang dituju 2. Pengguna melakukan klik kiri pada <i>marker</i> 3. Pengguna mengklik tombol data hari ini |
| Hasil Yang Diharapkan | Pada bagian diagram garis jumlah ujaran negatif, akan menampilkan informasi mengenai seluruh data ujaran negatif yang tersimpan di dalam database berdasarkan tanggal saat pengguna mengakses sistem yang divisualisasikan dengan menggunakan diagram garis |
| Hasil | Menampilkan informasi mengenai seluruh data ujaran negatif yang tersimpan di dalam database berdasarkan tanggal saat pengguna mengakses sistem yang divisualisasikan dengan menggunakan diagram garis |
| Status | Valid |

1.3.1.8 Pengujian Validasi Melihat Detail Data Kemarin

Tabel 0.18 Kasus Uji Melihat Detail Data Kemarin

| | |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | Melihat Detail Data kemarin |
| Prosedur | <ol style="list-style-type: none"> 1. Pada maps telah terdapat <i>marker</i> pada lokasi yang dituju 2. Pengguna melakukan klik kiri pada <i>marker</i> 3. Pengguna mengklik tombol data 1 hari lalu |

| | |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Hasil Yang Diharapkan | Pada bagian diagram garis jumlah ujaran negatif, akan menampilkan informasi mengenai seluruh data ujaran negatif yang tersimpan di dalam database pada satu hari sebelumnya saat pengguna mengakses sistem yang divisualisasikan dengan menggunakan diagram garis |
| Hasil | Menampilkan informasi mengenai seluruh data ujaran negatif yang tersimpan di dalam database pada satu hari sebelumnya saat pengguna mengakses sistem yang divisualisasikan dengan menggunakan diagram garis |
| Status | Valid |

1.3.1.9 Pengujian Validasi *Export Data Ke Excel*

Tabel 0.19 Kasus Uji *Export Data ke Excel*

| | |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | Export <i>data ke excel</i> |
| Prosedur | <ol style="list-style-type: none"> 1. Pengguna memilih menu dengan <i>icon save</i> pada navbar 2. Pengguna mengklik <i>export data ke xls</i> |
| Hasil Yang Diharapkan | Browser akan mengunduh dan menyimpan file <i>workbook</i> dengan ekstensi .xls |
| Hasil | Browser akan mengunduh dan menyimpan file <i>workbook</i> dengan ekstensi .xls yang berisi seluruh data ujaran negatif, positif, serta netral yang tersimpan di dalam database |
| Status | Valid |

1.3.1.10 Pengujian Validasi *Export Data Ke Excel Pada Kota Pilihan*Tabel 0.20 Kasus Uji *Export Data Ke Excel Pada Kota Pilihan*

| | |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | <i>Export Data Ke Excel Pada Kota Pilihan</i> |
| Prosedur | <ol style="list-style-type: none"> 1. Browser telah meload halaman home 2. Pengguna memilih icon save pada kota yang tersedia di dalam bagian data tabel save kota pilihan ke excel |
| Hasil Yang Diharapkan | Browser akan mengunduh dan menyimpan file <i>workbook</i> dengan ekstensi .xls yang berisi data ujaran negatif, positif, serta netral pada kota yang dipilih pengguna |
| Hasil | Browser mengunduh dan menyimpan file <i>workbook</i> dengan ekstensi .xls yang berisi data ujaran negatif, positif, serta netral pada kota yang dipilih pengguna |
| Status | Valid |

1.3.1.11 Pengujian Validasi Melihat Waktu Ujaran Negatif Terbanyak

Tabel 0.21 Kasus Uji Melihat waktu Ujaran Negatif Terbanyak

| | |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | Melihat waktu ujaran negatif terbanyak |
| Prosedur | <ol style="list-style-type: none"> 1. Browser telah meload halaman home 2. Pengguna melakukan klik kiri pada <i>marker</i> |
| Hasil Yang Diharapkan | Sistem akan menampilkan tiga waktu data dengan ujaran negatif terbanyak serta divisualisasikan dengan menggunakan diagram pie |
| Hasil | Sistem akan menampilkan tiga waktu data dengan ujaran negatif terbanyak serta divisualisasikan dengan menggunakan diagram pie |

| | |
|--------|-------|
| Status | Valid |
|--------|-------|

1.3.1.12 Pengujian Validasi Melihat Kota Ujaran Negatif Terbanyak

Tabel 0.22 Kasus Uji Melihat Kota Ujaran Negatif Terbanyak

| | |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | Melihat kota ujaran negatif terbanyak |
| Prosedur | <ol style="list-style-type: none"> 1. Browser telah meload halaman home 2. Pengguna memilih tombol update pada bagian rangking kota ujaran negatif terbanyak |
| Hasil Yang Diharapkan | Pada halaman home di bagian rangking kota akan menampilkan urutan kota dengan ujaran negatif tertinggi yang divisualisasikan dengan menggunakan diagram garis |
| Hasil | Pada halaman home di bagian rangking kota akan menampilkan urutan kota dengan ujaran negatif tertinggi yang divisualisasikan dengan menggunakan diagram garis |
| Status | Valid |

1.3.1.13 Pengujian Validasi Menghentikan Pengambilan Data

Tabel 0.23 Kasus Uji Menghentikan Pengambilan Data

| | |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Nama Kasus Uji | Menghentikan Pengambilan Data |
| Prosedur | <ol style="list-style-type: none"> 1. Pengguna telah membuka halaman home dan sebelumnya telah terdapat proses pengambilan data pada lokasi tertentu 2. Pengguna melakukan klik kanan pada <i>marker</i> |
| Hasil Yang Diharapkan | Proses pengambilan data akan berhenti yang ditunjukkan dengan hilangnya marker pada lokasi yang dipilih tersebut |

| | |
|---------------|------------------------------------------------------------------|
| Hasil | Proses pengambilan data berhenti serta marker terhapus dari maps |
| Status | Valid |

1.3.2 Kebutuhan Non Fungsional

1.3.2.1 Pengujian Validasi *Compability*

Pada pengujian validasi untuk compability dilakukan dengan menguji sistem yang dibangun dengan menggunakan browser yang beragam, dan browser yang umum digunakan oleh pengguna dalam menjalankan aplikasi berbasis web. Browser yang digunakan dalam pengujian ini yaitu Mozilla firefox, dan google chrome. Hasil pengujian compability dapat dilihat pada Tabel Kasus uji Compability.

Tabel 0.24 Pengujian Validasi *Compability*

| NO | Jenis Browser yang digunakan | Status |
|-----------|-------------------------------------|---------------|
| 1. | Google Chrome | Valid |
| 2. | Mozilla Firefox | Valid |

DAFTAR PUSTAKA

- Al-garadi, M. A., Varathan, K. D. & Ravana, S. D., 2016. Cybercrime detection in online communications: The experimental. *Computers in Human Behavior*, Issue 63, pp. 433-443.
- Anon., 2014. www.apjii.or.id. [Online] [Accessed September 2017].
- Anon., 2015. *Dinas Kesehatan kabupaten Lamongan*. [Online] Available at: <https://lamongankab.go.id/dinkes/cyberbullying-amankah-buah-hati-anda/> [Accessed 2017].
- Arief, 2012. *Artikel Teknik Informatika dan Sistem Informasi*. [Online] Available at: <http://informatika.web.id/sequence-diagram.htm> [Accessed 7 Oktober 2017].
- Besley, 2009. *Cyberbullying Research Center*. [Online] Available at: <http://www.cyberbullying.org/> [Accessed 2018].
- Carley, K. M. et al., 2015. *Twitter Usage in Indonesia*, Pittsburgh: Center for the Computational Analysis of Social and Organizational Systems.
- Cassidy, W., Faucher, C. & Jackson, M., 2013. *Cyberbullying among youth: A comprehensive review of current international research and its implications and application to policy and practice*. *School Psychology International*, 34(6), pp. 576-612.
- Dehue, F., Völlink, T. & Bolman, C., 2008. *Cyberbullying: Youngsters' Experiences and Parental Perception*. *CyberPsychology & Behavior*.
- Dredge, R., JohnGleeson & Garcia, X. I. P., 2014. *Cyberbullying in social networking sites: An adolescent victim's perspective*. *Computers in Human Behavior*, Volume 36, pp. 13-20.
- Fowler, M., 2004. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. 3th ed. Boston: Addison-Wesley Professional.
- Hariyanti, E., 2008. *Pengembangan Metodologi Pembangunan Information Dashboard*.
- Hidayat, A. N., 2015. Analisis Sentimen Terhadap Wacana Politik Pada. *Jurnal Elektronik Sistim Informasi Dan Komputer (JESIK)*, 1(1).
- Januarita, D. & Dirgahayu, T., 2015. *Pengembangan Dashboard Information System (DIS) Studi Kasus : Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) di. Jurnal Infotel*, 7(2).
- Kusuma,, M. E. & Budisusanto, Y., 2015. *Aplikasi Google Maps Api Dalam Pengembangan Sistem Informasi Geografis (Sig) Pariwisata Berbasis Web*.

- Mahdia, F. & Noviyanto, F., 2013. Pemanfaatan Google Maps Api Untuk Pembangunan Sistem Informasi Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis Mobile Web. *Jurnal Sarjana Teknik Informatika*, 1(1).
- Makice, K., 2009. *Twitter API: Up and Running*. s.l.:s.n.
- Napitupulu, E. L., 2012. *kompas.com*. [Online] Available at: <https://edukasi.kompas.com/read/2012/01/23/18024424/Kekerasan.di.Dunia.Maya.Mengancam.Anak-anak> [Accessed 9 April 2018].
- Parlindungan, M. & Sapto, A., 2015. *Profil Pengguna Internet Indonesia 2014*. Pertama, Maret 2015 ed. Jakarta: Asosiasi Penyelenggara Jasa.
- Pearce, K. E. & Rice, R. E., 2017. Somewhat Separate and Unequal: Digital Divides, Social Networking Sites, and Capital-Enhancing Activities. *SAGE journals*.
- Poblete, B., Garcia, R., Mendoza, M. & Jaimes, A., 2011. Do all birds tweet the same?: characterizing twitter around the world. *Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 1025-1030.
- Price, M. & Dalglish, J., 2010. *Cyberbullying: Experiences, Impacts and Coping Strategies as Described by Australian Young People*. *Youth Studies Australia*, 29(2), pp. 51-59.
- Rahayu, F. S., 2012. *Cyberbullying Sebagai Dampak Negatif Penggunaan Teknologi*. *Jurnal Sistem Informasi (Journal of Information System)*, 8(1), pp. 22-31.
- Rozi, I. F., Pramono, S. H. & Dahlan, E. A., 2012. Implementasi Opinion Mining (Analisis. *Jurnal EECCIS*, 6(1).
- Sommerville, I., 2011. *Software Engineering (9th Edition)*. 9th ed. Boston: Addison-Wesley.
- Svennerberg, G., 2010. *Beginning Google Maps API 3*. s.l.:Apress.
- Utami, Y. C., 2014. *Cyberbullying di Kalangan Remaja (Studi tentang Korban Cyberbullying di Kalangan Remaja di Surabaya)*. *Journal Universitas Airlangga*, Volume 3, pp. 1-10.